DOCUMENT RESUME

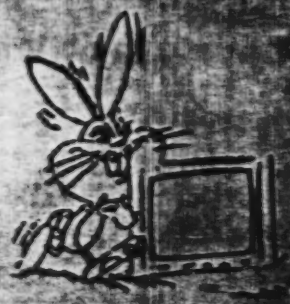
ED 268 964 IR 012 022

AUTHOR	Amburgey, Valeria, Ed.; Olivier, Terry A., Ed.
TITLE	Proceedings for the 4th Annual Micros on Parade Conference (4th, Houston, Texas, June 7-8, 1985).
INSTITUTION	Education Service Center Region 4, Houston, Tex.
PUB DATE	Jun 85
NOTE	117p.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Guides - Classroom Use - Guides (For Teachers) (052) -- Computer Programs (101)
EDRS PRICE	MF01/PC05 Plus Postage.
DESCRIPTORS	Community Education; *Computer Assisted Instruction; *Computer Graphics; *Computer Literacy; *Computer Software; Elementary Secondary Education; *Microcomputers; Program Descriptions; *Programing; Programing Languages; Teaching Methods
IDENTIFIERS	LOGO Programing Language

ABSTRACT

This document contains 25 presentations on five broad topics: the interface of computers with instruction; computer applications; computer graphics; computer programming; and general interest sessions. A foreword by Dr. Valeria Amburgey of Sam Houston State University precedes the following papers: (1) "Fourth and Fifth Grade Computer Centers Soliciting & Utilizing Computer Volunteers" (Susan Wethington); (2) "Apples in the Classroom" (outline only, Dorothy MacDonald and Joann Nelson); (3) "Implementing LOGO in the Elementary Computer Lab" (Joan Brummett); (4) "LOGO--Graphics and Beyond" (Frank Matthews); (5) "LOGO Graphics Work in Apple LOGO, and Can Be Saved on Video with Sound Added" (Peggy Hill); (6) "Story Grammar at the Computer: Models and Process" (Peggy Hill); (7) "Competitive Math Teaching Strategies Using the Computer" (Gaye Russell); (8) "Side Stepping the Essential Elements" (Dena Bankston); (9) "Computer Literacy for the Small Community" (Pam Wood); (10) "Trials and Tribulations of Computer Literacy" (handout, Rita Depree); (11) "Word Processing: A Tool for Learning" (Martha Lawrence); (12) "Welcome to the Big Apple! Appleworks" (Pamela Cho); (13) "Understanding Database Management" (Terry A. Olivier); (14) "Graphic Software: Which Ones to Purchase and How to Use Them" (Gerald Pollard); (15) "Introducing PASCAL through Karel the Robot" (Dell Simon); (16) "Using LOGO to Program Karel" (Nancy Siragusa); (17) "Manipulating the Character Generator" (Betty Carter, Barbara Edwards, and Ken Ismert); (18) "Manipulating Memory on the Apple II Computer" (Jeffery W. Bloom); (19) "Interactive Video and Graphic Special Effects" (Lois E. Gordon Mosely); (20) "Effective Use of Computers in the Science Classroom" (Jeffery W. Bloom); (21) "Hands-On in the Science Classroom" (Nan Porter); (22) "The Rib Strikes Back: Countering Gender Inequities in Computer Use (Danna Fertsch and James R. Aman); (23) "The Home Connection" (Mark Sullivan); (24) "The Computer in the Christian Environment" (Bill Cobb); and (25) "Teaching Junior High Computer Literacy: The Texas Basic in the Summer" (Debbie Roberts). A list of reference books and computer magazines available through the Region IV Education Service Center is included. (THC)

ED268964



PROCEEDINGS FOR THE 4th Annual Conference

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

JUNE 7 and 8, 1985

*Graphics *Telecommunications

*Computer Literacy *Word Processing

*Instructional Applications

IR012022

Region IV Education Service Center

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Valeria Amburgey
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

PROCEEDINGS

FOR THE

4TH ANNUAL

MICROS ON PARADE

CONFERENCE

REGION IV SERVICE CENTER

HOUSTON, TEXAS

Editors:

Dr. Valeria Amburgey, Sam Houston State University
Huntsville, Texas

Dr. Terry A. Olivier, Emporia State University
Emporia, Kansas

IR013022

Conference Overview

The 4th Annual "Micros on Parade", a two day conference sponsored by the Region IV Education Service Center was held at the Region IV Service Center in Houston, Texas on June 7th and 8th. Over 250 teachers and administrators from the Region IV area took part in the conference.

A wide degree of interest in the uses of microcomputers was reflected in the variety of topics presented. This document contains the papers which were submitted for inclusion into the proceedings. The papers can be divided into the following 5 broad topics: 1) The Interface of Computers with Instruction, 2) Computer Applications, 3) Computer Graphics, 4) Computer Programming, and 5) General Interest Sessions.

TABLE OF CONTENTS

PAGE

FORWARD
Dr. Valeria Amburgey, Sam Houston State i

FOURTH AND FIFTH GRADE COMPUTER CENTERS SOLICITING & UTILIZING COMPUTER VOLUNTEERS
Susan Wethington, Houston ISD 1

APPLES IN THE CLASSROOM
Dorothy MacDonald & Joann Nelson, Angleton ISD .. 5

IMPLEMENTING LOGO IN THE ELEMENTARY COMPUTER LAB
Joan Brummett, Fort Bend ISD 7

LOGO -- GRAPHICS AND BEYOND
Dr. Frank Matthews, University of Houston-C.L. .. 12

LOGO GRAPHICS WORK IN APPLE LOGO, AND CAN BE SAVED ON VIDEO WITH SOUND ADDED
Peggy Hill, University of Houston-U.P. 14

STORY GRAMMAR AT THE COMPUTER: MODELS AND PROCESS
Peggy Hill, University of Houston-U.P. 22

COMPETITIVE MATH TEACHING STRATEGIES USING THE COMPUTER
Gaye Russell, Alief ISD 26

SIDE STEPPING THE ESSENTIAL ELEMENTS
Dena Bankston, Cypress-Fairbanks ISD 29

COMPUTER LITERACY FOR THE SMALL COMMUNITY
Pam Wood, Anahuac ISD 32

TRIALS AND TRIBULATIONS OF COMPUTER LITERACY
Rita Depree, North Forest ISD 35

WORD PROCESSING: A TOOL FOR LEARNING
Martha Lawrence, La Marque ISD 36

WELCOME TO THE BIG APPLE! APPLEWORKS
Pamela Cho, Galveston ISD 40

UNDERSTANDING DATABASE MANAGEMENT
Dr. Terry A. Olivier, Emporia State University .. 50

GRAPHIC SOFTWARE: WHICH ONES TO PURCHASE AND HOW TO USE THEM
Gerald Pollard, Texas School for the Deaf 53

INTRODUCING PASCAL THROUGH KAREL THE ROBOT
Dr. Dell Simon, University of St. Thomas 58

USING LOGO TO PROGRAM KAREL
Nancy Siragusa, St. Michael School 62

MANIPULATING THE CHARACTER GENERATOR
Betty Carter, Barbara Edwards, & Ken Ismert
Spring Branch ISD & University of Houston-U.P. .. 70

MANIPULATING MEMORY ON THE APPLE II COMPUTER
Jeffery W. Bloom, University of Houston- U.P. ... 73

INTERACTIVE VIDEO AND GRAPHIC SPECIAL EFFECTS
Lois E. Gordon Mosely, Houston ISD 77

EFFECTIVE USE OF COMPUTERS IN THE SCIENCE CLASSROOM
Jeffery W. Bloom, University of Houston-U.P. 78

HANDS-ON IN THE SCIENCE CLASSROOM
Dr. Nan Porter, Kinkaid School 82

THE RIB STRIKES BACK: COUNTERING GENDER INEQUITIES IN COMPUTER USE
Dr. Danna Fertsch & James R. Aman
University of Houston-U.P. & ST. Pius School 90

THE HOME CONNECTION
Mark Sullivan, La Marque ISD 100

THE COMPUTER IN THE CHRISTIAN ENVIRONMENT
Bill Cobb, Second Baptist School 101

TEACHING JUNIOR HIGH COMPUTER LITERACY
THE TEXAS BASIC IN THE SUMMER
Debbie Roberts, Lamar Consolidated ISD 106

FORWARD

The complexion of the classrooms of today differ greatly from the classrooms of thirty, twenty, or even ten years ago. The dictionaries, encyclopedias, textbooks, chalk and erasers can still be found, but computers have also found a place in the modern day educational environment. This emergence of the computers into the classrooms may not have occurred overnight, but it did occur quickly and often without planning. The educational community quickly realized that this haphazard approach to instructional computing was inappropriate. Educators everywhere began developing guidelines, definitions, and levels of hierarchial skills for instructional computing. Legislatures also became involved by establishing requirements for all school districts to follow. Texas' Chapter 75 curriculum with the inclusion of computer-related courses, such as computer literacy, computer mathematics, and computer science, is a direct result of the movement for an organized approach to instructional applications of computers. Many of the sessions you will attend at this conference will reflect the thoughts and ideas of these educators and legislative bodies.

Computer literacy, as a course offering, will be implemented in school districts across the state of Texas as part of the Chapter 75 requirements. Students in junior high school will be required to participate in a one semester course or otherwise show mastery of the essential elements identified below. Local district policy will determine if the course will be a seventh or eighth grade requirement and district officials may also identify additional requirements or time frames for local credit. The five essential elements for computer literacy as outlined in Chapter 75 are:

1. Computer-related terminology and use.
2. History and development of computers.
3. Use of the computer as a tool.
4. Communicating instructions to the computer.
5. Problems and issues of computer use in the society.

Chapter 75 also includes descriptions of a number of secondary level computer courses which school districts may offer. Students in grades nine through twelve may elect to pursue an advanced high school program as opposed to the regular program. Of the 22 units required for the advanced program, students will be required to complete one unit of computer science. This computer science credit may be selected from:

1. Computer Mathematics I or II
2. Business Data Processing and Introduction to Computer Programming
3. Vocational Data Processing

4. Vocational Computer Programming
5. Business or Vocational Word Processing
6. Computer Science I or II

The selection of Computer Mathematics as the computer science credit is permissible as long as it does not also count towards the necessary math credits. Notice also that a distinction has been made between Computer Mathematics and Computer Science. The essential elements for these two courses do represent different philosophies and content.

Computer literacy and computer science, as described in the preceding paragraphs, identify specific areas of direct instruction in computing and will be a primary focus in a few of the sessions at this conference. These courses will focus on the computer as an object of instruction rather than as a tool to aid in other classroom activities. Teachers and administrators are encouraged, however, to use computers whenever appropriate and possible for other classroom applications. Computer-assisted instruction (CAI) or computer-managed instruction (CMI) are other examples of computer use in the classroom.

CAI is a means of providing instruction in the content areas to students via the computer. CMI provides instruction to students with the additional capabilities of the computer to store information about the students (scores, responses, material covered, etc.) for the teachers' retrieval at a more convenient time. The instruction provided in CAI or CMI may contain the characteristics of one or more of the following instructional techniques:

DRILL AND PRACTICE: The computer becomes an electronic flashcard by providing questions to the student, allowing the student to respond, and evaluating the student's response to be correct or incorrect.

TUTORIAL: The student interacts with the computer to 'learn' content which may be new to him or her. Responses from the student may be required for the continuation of the program. Responses are often evaluated and program continuation is based upon the type of response received.

SIMULATION: Situations which may be difficult to simulate within a classroom setting can often be simulated on a computer. Students can become role players in the situation depicted by selecting courses of action to be pursued via the computer.

EDUCATIONAL GAME: Content can be introduced or reinforced in the context of a game via the computer.

PROBLEM SOLVING: This type of program will encourage the student to examine the information given, develop strategies for a successful conclusion, and test the strategies for correctness.

Speakers in the sessions you attend may refer to CAI, CMI, or another type of instructional application of the computer in the content areas. The definitions above may prove beneficial to those relatively new to instructional computing.

Novices in instructional computing may also find the following terms and definitions helpful while attending the variety of sessions at this conference. At times newcomers to the world of instructional computing are overwhelmed by the impression of a strange language being utilized. The list below may serve as a primer to the world of instructional computing.

BASIC: An acronym for Beginner's All-purpose Symbolic Instruction Code. A programming language developed at Dartmouth College in the 1960s.

BUG: An error in a program that does not allow the computer to complete a successful execution of the program.

CPU: Central processing unit which contains the memory, arithmetic and control units.

CRT: A type of TV screen, known as a cathode ray tube, connected to a computer.

CURSOR: A symbol which appears on the screen at the location where the next character entered will appear.

DISK DRIVE: A piece of hardware which records and retieves data from disks.

DISK: A storage medium suitable for storing magnetically recorded data.

HARD COPY: A paper copy of the output.

HARDWARE: The physical components of a computer system.

INPUT: Information entered into a computer from an outside source, i.e. keyboard, disk drive, etc.

INITIALIZE: A method of preparing a disk for future storage and retrival of data. FORMAT is another term often used in the same context.

LOGO: A structured language developed as a result of Dr. Seymour Papert's work at MIT. Most well-known for its turtle graphic capabilities.

PASCAL: A structured language which was named for Blaise Pascal.

MEMORY: Locations inside the computer which store data and programs.

OUTPUT: Information received from a computer via an output device such as a CRT or printer.

PROGRAM: A sequence of instructions which direct the computer to perform certain tasks.

SOFTWARE: The computer programs.

Current legislation pertaining to computer education in Texas and definitions of computer-related terminology have been discussed in the preceding paragraphs in an effort to provide a bridge from here to the conference sessions and the papers contained within this book. This conference has been designed as a means by which area educators can share their experiences and expertise with others. Participants are encouraged to inquire, explore, and also share their thoughts and ideas as well. And, it is hoped that the experiences and information provided during this conference will form the basis of new and rewarding instructional computing applications in your future.

Dr. Valeria Amburgey
Sam Houston State University
Huntsville, Texas

FOURTH AND FIFTH GRADE COMPUTER CENTERS
SOLICITING & UTILIZING COMPUTER VOLUNTEERS
Presenter: Susan Wethington

Point 1: Centers

The computer revolution is genuinely designed to offer something a bit different than traditional methods and outcomes. It is acclaimed as having the character or ability of meeting individual needs. It has an innate ability to motivate, which is noted by even unbelieving educators. In addition to dealing with individual needs, the computer situation is supposed to almost run itself. Are these things really possible? At what point will there be accountability for computer achievements?

There are many unanswered questions about appropriate computer make-up and delivery. Individualized computer centers in a lab setting managed by a teacher is a possible approach.

Point 2: Volunteers

Another important computer consideration is resources. Who is more interested in the education of students than the parents of the children we teach? They are definitely a positive resource to consider in a computer situation. It is very difficult to help 20-30 students all at one time, for each student may be having a different computer problem simultaneously.

Many parents are eager to learn about computers, so often they are willing to do volunteer work in return for a small amount of training.

OVERVIEW OF THE SESSION

I. Why centers?
   A. Definition of centers

II. How to set up centers.
   A. Geographic centers vs. Flexible centers
   B. Yearly or Unit assignments
   C. Instruction types
      1. Teacher directed sessions
      2. Written direction instructon
      3. What activities belong in centers?
      4. Scheduling and time slots.

III. Accountability
   A. Evaluation Forms
   B. Timing

IV. Volunteers
   A. How to solicit
   B. Who to solicit
   C. Various methods of utilizing volunteers

ORGANIZING CENTERS
Presenter: Susan Wethington

I. Look at what materials you already have. There are some inexpensive materials you can make.
NOTES: 1. Flwchart Folder-
2. Computer Library-
3. Project Suggestions-
4. Word Processing Cards-
5. Programming Task Cards-
6. Calculator Activities-
7. Newspaper
8.
9.

II. Define Major Topic Areas
NOTES: 1. Literacy-
2. Programming-
3. Typing/Word Processing-
4. Drill & Practice on Content Areas-
5. Problem Solving Software-
6.
7.

III. Grouping Activities
NOTES: LITERACY
programming films
flowcharts
articles
project
PROGRAMMING
Basic
Logo
WORD PROCESSING
newpaper
letter

IV. Instruction Folders & Long Term Assignments
NOTES: 1. Write instructions for a computer related assignment-

2. Write 3 simple computer assignments-
a.
b.
c.
3. Suggestions:
a. Have a sheet for each student to record daily what they did that day.
b. Have a Master Score Sheet for students to record their own scores.
c. A computer folder is a good idea.

EXAMPLE SHEET
Presenter: Susan Wethington

I. Grouping Example

1	2	3	4	
SAM	JON	ROB	PEG	
PHIL	JILL	SAL	LYNN	
SUE	DEB	KIM	KELI	CLASS 5A
CARI	AL	JUNE	KEN	
VAN	DORI	ANN	JOY	
TERI	PAM	JERI	JIM	

II. Self-Made Materials Examples

**FLOWCHART QUESTION: Write a flowchart to represent the following situation. I like to watch the 3:00 movie. If I get home by 3:00, I watch the movie. If I do not get home by 3:00, I don't watch it.

**COMPUTER LIBRARY: Have students collect magazine & newspaper articles about computers. Have them do a certain number of Article Log Entries as follows:
Article----------M/Yr--------Categories---------- ----------
Main Idea (2 sentence max.)----------
**PROJECT SUGGESTIONS: Make a model of a computerized robot...Make a scrapbook of computer articles w/ short descriptions...compare different brands of computer by price and ability.....

**TASK CARDS (LOGO, BASIC, WORD PROCESSING respectively): Define and SAVE a square...Make the picture that you see on this task card using Logo, and add something to it....Write a Basic program that tells your name and address....Write another one that asks the user his name & age....Write a letter home on the word processor...Also, write a creative computer story on the word processor...

III. Organizational Forms

**ASSIGNMENT SHEET:
...LITERACY...Write a report about computers...Do ten flowchart situations...Do one computer project...etc.
...PROGRAMMING...Do 3 Basic programs...Do 2 Logo programs...Watch 2 programming films...etc.
...WORD PROCESSING...Write a letter home...Write a creative story about computers and print it out...etc.

**INSTRUCTION FOLDER:
...PROGRAMMING FILMS..1 You may have a partner. 2 Go through the films in order. 3 Take the test over the film. 4 Write the answers in your folder. 5 Let someone in your group check your answers. 6 The answer keys are on my desk. 7 If you get less than 80 do the film again.

**MASTER SCORE SHEET:
...Parts of Speech..Which? N V ADV ADJ..Section: a b c d..correct---
...States & Capitals..date ( )French or Spanish..Score ( )

**CHECKLIST or DATELOG or DIARY
...LITERACY 1 Films----/----/----... 2 Flowcharts----/----/---... 3
Project ----/----/----
...PROGRAMMING 1 Basic----/----/----/... 2 Logo----/----/----
...WORD PROCESSING 1 Letter----/----/----/... 2
Newspaper----/----/----

****

COMPUTER VOLUNTEERS

I. How to solicit volunteers & approach training.
Volunteers can sign up at a Parent Teacher Meeting. Also, it is a good idea to write a letter explaining the tasks YOU want them to do for you.

The letter should contain: a list of the tasks or jobs (Classroom assistant, Logo extention, Kindergarten Computer Assistant, Newpaper, Surveys, Interviews...), descriptions of those jobs, emphasize the importance of the tasks, possible time slots...(Mon. 9:00, Tues. 11:00, Wed. 9:30

Offer an inservice to train the volunteers...Make a folder for each volunteer SPELLING OUT EVERY TASK AND PROCEDURES YOU WANT THEM TO DO...

II. Who to use as a volunteer.
It is best to ask other teachers etc. if they know any of your new volunteers. Give new unknown volunteers only small short-term tasks in order to check their ability and compatibility with you.

III. Utilizing volunteers
Class computer aids, Special Interest Groups, Kindergarten Parents

BEST COPY AVAILABLE

APPLES IN THE CLASSROOM

Dorothy Mac Donald
Joann Nelson

TASK FORCE FOR TECHNOLOGY - October 1983

SCOPE AND SEQUENCE FOR ANGLETON I.S.D. (Attached)

THIRD GRADE

Children at work on the computer

Keyboarding by Merit
Master Type by Scarborough
Math by Milliken

Two computers on third grade level
Rotated one day in each classroom

FOURTH GRADE

Children at work on the computer

Cloze by Milliken
Comprehension Power by Milliken
Tutorial Comprehension by Random House

Graphics in the classroom

Robots Unit
Follow-up: Creative writing
History of robots

Two computers on fourth grade level
Rotated one day in each classroom

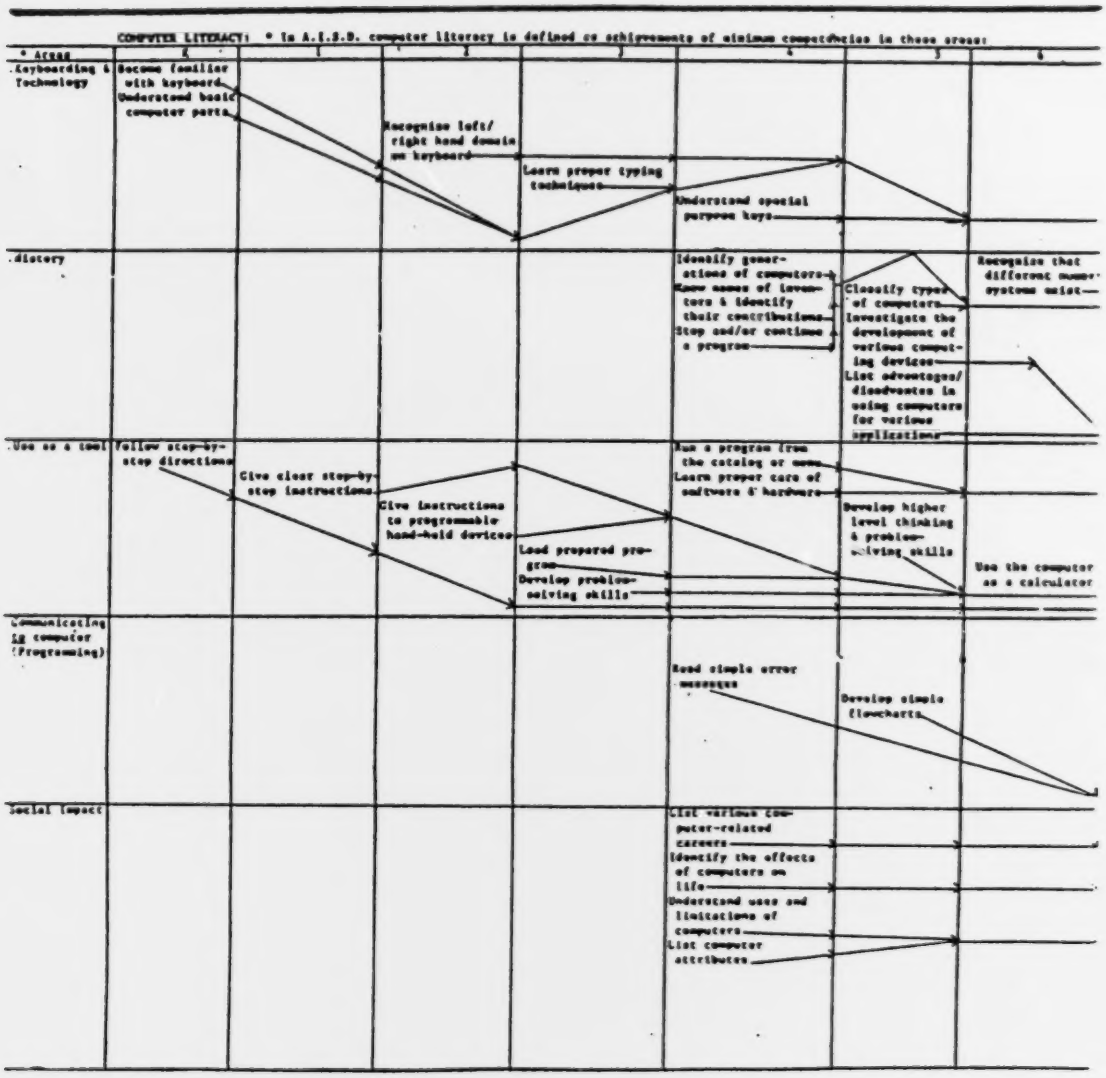
SUMMER PROGRAMS

Hands-on training for teachers on the computer
Computer Camp
Beginners - Low resolution graphics
Intermediates - Continuing low resolution and high resolution graphics
Pilot Program - Keyboarding for Jr. High and High School

BUDGET

BEST COPY AVAILABLE

SCOPE AND SEQUENCE
ANGLETON I. S. D.


COMPUTER LITERACY: • In A.I.S.D. computer literacy is defined as achievements of minimum competencies in these areas:
• Areas
K
1
2
3
4
5
6
.Keyboarding & Technology
Become familiar with keyboard. Understand basic computer parts.
Recognize left/right hand domain on keyboard
Learn proper typing techniques
Understand special purpose keys
.History
Identify generations of computers. Know names of inventors & identify their contributions. Stop and/or continue a program
Classify types of computers. Investigate the development of various computing devices. List advantages/disadvantages in using computers for various applications
Recognize that different number systems exist
.Use as a tool
Follow step-by-step directions
Give clear step-by-step instructions
Give instructions to programmable hand-held devices
Load prepared programs. Develop problem-solving skills
Run a program from the catalog or menu. Learn proper care of software & hardware
Develop higher level thinking & problem-solving skills
Use the computer as a calculator
Communicating w computer (Programming)
Read simple error messages
Develop simple flowcharts
Social Impact
List various computer-related careers
Identify the effects of computers on life
Understand uses and limitations of computers
List computer attributes

IMPLEMENTING LOGO IN THE ELEMENTARY COMPUTER LAB

by Joan Brummett
Townewest Elementary Building Computer Liaison
Fort Bend Independent School District

Why Logo?

Computer Assisted Instruction (CAI) has a valid place in today's computer labs. It has proven itself useful with all levels of students and in many subject areas. However, as useful as it is, it does not encourage the child to expand his thinking skills beyond the parameters of the CAI program itself. Seymour Papert, who developed Logo, felt that it is important that students, not only learn to use computers, but that they also learn something about the thinking process itself. The Logo language is designed to accomplish this and it is hoped that these thinking processes will spill over into all academic areas. Papert followed Piagetian ideas of child development in creating the Logo language. A child formulates a theory, then tests and revises that theory.

When should Logo be introduced?

In the Fort Bend District, we have chosen to begin Logo in the third grade. In general, third grade students have enough maturity and knowledge to make use of the concepts introduced in Logo in the lab setting. There are simplified versions of Logo which would be suitable for younger students.

Fort Bend's Implementation Program

Fort Bend began a pilot program early in 1983 to determine which computer to purchase for their elementary labs. Extensive teacher training (discussed later) was begun. We opened the computer labs during the 1983-1984 school year. A commitment was made to use the computers in the areas of reading, language arts, math and study skills. A starter set of CAI software, selected by a group of third, fourth, and fifth grade teachers was provided at each campus to address these curriculum areas. At this stage, Logo was used on a limited basis. The labs were open to fourth and fifth graders. The CAI work gave students an opportunity to become familiar with the computer before any programming was introduced. Although CAI still has a major roll in the Elementary Labs, the remainder of this paper will stress what the district has done with Logo. The use of Logo became districtwide during the 1984-1985 school year. Third

graders were also now included and the Elementary Labs were complete.

The district made an initial purchase of one computer system per 80 students served. A computer system consisted of one Apple IIe computer, one disk drive, and one color monitor. In addition, each lab was provided a printer. Since that time, PTA/PTO groups have purchased many additional systems for their schools. The ideal we are aiming for is a minimum of one system per every 40 students.

Students are scheduled for hands-on activities a minimum of twenty minutes, twice a week--two students per computer. This varies from school to school depending on the number of computers. We do not have a Computer Lab teacher, therefore, each teacher is responsible for taking her own class to the lab.

The Important Key---Training

Good teacher training is the heart of a successful Logo program. Three major elements should be included in the training program. First, of course, the teachers should be taught to use the program itself. Secondly, the teachers should be taught how to teach Logo. Many educators are not comfortable with teaching using the Logo discovery method. It is necessary that teachers realize why the method is important and see the method modeled for them by the teacher-trainer. Educators should be encouraged to take the role of initiator, then step back and let the children set their own pace and direction. The teacher in a Logo lab should not be there to provide answers, but to guide children to find their own answers and seek new questions.

The next element of teacher training is to give teachers help in sequencing the Logo concepts. Although this may vary once the children actually get started, teachers need a foundation to build on. This will go a long way toward building teacher confidence.

Fort Bend, with 17 elementary schools and approximately 255 teachers involved in the computer labs by next year, has an extensive training program. The basic fifteen hour, three part program was developed by our Instructional Computer Specialist, Dede Heidt. Teachers within the district are used as trainers. The first part introduces the teacher to the basics of how to use a computer. The second part gives the teacher an opportunity to become familiar with some of the CAI software available. The third part, a five hour course, is designed to prepare the teacher for teaching Logo. Because teachers felt they needed more

Logo training, once they actually began teaching, an additional seven hour Advanced Logo was instigated. Other inservices will be added as the need arises.

A Sample Beginning Logo Curriculum

In the summer of 1984, interested teachers met to develop the Logo Activity Guide, a manual presenting a logical sequence of concepts to be taught in Third, Fourth, and Fifth grades. The teachers coordinated the guides with the objectives in new Texas curriculum. We have implemented the concepts presented in the Third grade guide on all three levels this year. To give an idea of the sequence followed, a brief description of the contents of this guide is presented below.

1) Learning to use the keyboard itself.
2) Learning to access Logo from the disk.
3) Learning to use basic Turtle Commands: FORWARD, BACKWARD, RIGHT, LEFT, CLEARSCREEN and HOME. Students can now make simple rectangular shapes and move the turtle through transparent mazes which are taped to the monitor screen.
4) Learning to understand and correct errors. The use of the arrow keys is introduced.
5) Learning to use the basic commands in more complicated maze patterns. Working with angles and degrees of a circle are addressed.
6) Learning to use the various angles needed in formulating shapes such as the triangle. The students will discover that the Turtle must always move 360 degrees to form a shape.
7) Learning Logo commands PENUP, PENDOWN, AND PENERASE. This allows students more freedom in making designs and correcting mistakes.
8) Learning Logo commands SETPC(set pen color), SETBG(set background color), and HIDETURTLE. This adds a whole new creative element to the children's drawings.
9) Learning the commands TEXTSCREEN, FULLSCREEN, SPLITSCREEN, CONTROL-T, CONTROL-L AND CONTROL-S. These commands add variety to their drawings.
10) Learning the commands to add, subtract, multiply, and divide.

This sequence may need to be adjusted to meet the needs of individual students and classes. Once these skills are mastered, students are ready to move on to the more challenging skills of the Edit mode, saving programs on the disk, recursion, etc., presented in the Fourth and Fifth Grade guides.

RESOURCES

Fort Bend Independent School District (1984). Logo activity guide. Sugar Land, Texas.

Fort Bend Independent School District (1984). Taking the first 'byte' from your Apple--beginning Logo. Sugar Land, Texas.

Minnesoto Educational Computing Consortium (1983). Apple Logo in the classroom. St. Paul, Minnesota.

Minnesoto Educational Computing Consortium (1983). Introduction to Logo for teachers. St. Paul, Minnesota.

Papert, S. (1980). Mindstorms. New York: Basic Books, Inc.

Riordon, T. Creating a Logo Environment. Eugene, Oregon: Alpine Educational Computing Services.

Watt, D. (1984). Learning with Apple Logo. New York: McGraw-Hill Book Company.

Willis, J., Johnson, D., & Dixon, P. (1983). Computers, teaching, and learning. Beaverton, Oregon: Dilithium Press.

A special thanks to Dede Heidt, Instructional Computer Specialist, Fort Bend Independent School District, for her advice in preparing this paper.

LOGO -- GRAPHICS AND BEYOND

Frank F. Matthews
Associate Professor, UH-CL

The LOGO language has been in use for a number of years but has developed a reputation of being only a simple language for teaching graphics. This presentation will stress the complexity of the language, applications of graphics, and its use in non-graphics situations.

The LOGO language has been in use in the schools for a number of years. Unfortunately, because of its ease of access and the power and novelty of the turtle graphics implementation, it has come to be viewed as a simple language for use only with elementary school students and their teachers. For the past few years I have worked with the language and taught a variety of teachers. During that time I have developed an appreciation for the power and range of LOGO in a wide variety of situations. Today I will consider applications - including some using graphics -, list manipulations for AI and language arts, and mathematics applications. The point I would like you to consider is that LOGO is the appropriate language to use for the introduction of programming concepts and for the development of programming skills until a complex fully structured procedural language is needed. In other words LOGO then PASCAL (or FORTRAN or ADA) but no BASIC.

One simple way to use LOGO beyond teaching graphics programming is to use the powerful graphics capabilities to teach another subject area. One such example is that of map exercises in social studies. Here we define a set of procedures to move the cursor character according to specified directions. One such procedure will cause motion in a South by South-East direction for a variable number of units.

```
TO S'S'E :X
 SETH 202.5
 FD :X
END
```

Similar procedures are defined for each of the other directions. The ' characters are used to avoid interference with LOGO primitives. The student would then use these procedures to see the result of following a set of map directions. Different sets of procedures could be used for sets of 4, 8, or 16 directions. The exercises could be made more meaningful by including a random factor say -4 to 4 degrees in each motion to simulate compass error.

LOGO can also be used effectively to manipulate words and lists of words. A useful application is the short term memory drill to reinforce the learning of spelling words. This application can readily be implemented in LOGO as you can see from the procedures in the Appendix available from the author.

Similar and very effective applications can be developed in sentence generation and story telling. A number of authors present such applications such as Daniel Watt "Learning With Logo" Chapter 13 or Earl Babbie "Apple Logo for Teachers" Chapters 17-19.

One of the extremely powerful tools in the LOGO language is the ability to move at will from a defined procedure to a text list (TEXT) and from a suitable text list to a procedure (DEFINE). These tools are a part of the LISP language and are the key to applications in artificial intelligence. These let the program learn and modify itself based on the result of a trial or a series of trials.

Another powerful tool has a wide variety of applications in the area of mathematics. LOGO is a fully recursive language. This means that a procedure can call itself. This enables the effective presentation of a number of mathematical algorithms.

One such is Euclid's Algorithm for the computation of the GCF. The algorithm is simplicity itself.

Compute the remainder R of two numbers A & B where A>B

If R=Ø then the GCF is B
otherwise the GCF is the GCF of B & R.

This enters beautifully in LOGO. We will assume :A < :B

```
TO EUCLID :A :B
   MAKE "R REMAINDER :A :B
   IF  :R=Ø  [OUTPUT  :B]   [OUTPUT
EUCLID :B :R]
END
```

The example procedures in the Appendix display the output and arrange the input effectively but the operative part is as described above.

Still another application from mathematics is the computation of prime numbers. The following procedures will compute such prime numbers. You should note the recursive use of TESTP to test for divisibility of successive primes.

```
TO PRIME
 PRINT 2
 MAKE "PRIMES [2]
 MAKE "NUM 3
 REPEAT 499999 [TRY MAKE "NUM :NUM
+2]
END

TO TRY
 IF  TESTP :NUM :PRIMES THEN  PRINT
:NUM IF :NUM<1ØØØ THEN MAKE "PRIMES
(SENTENCE :PRIMES :NUM)
END

TO TESTP :N :P
 MAKE "X FIRST :P
 IF :N < :X * :X THEN OUTPUT 1=1
 IF REMAINDER :N :X =Ø THEN  OUTPUT
1=Ø
 OUTPUT TESTP :N BUTFIRST :P
END
```

Still other applications in the mathematics area can be made with the DEFINE primitive. An entered function can be the output of a procedure and the procedure X can output a number. These procedures can then be used to draw graphs, to create a function table, or even in numerical analysis algorithms. To see examples of such implementations see the appendix.

LOGO GRAPHICS WORK IN APPLE LOGO, AND CAN BE SAVED ON VIDEO WITH SOUND ADDED

by

Peggy Hill

Felesa Baker (Researcher)

Intended audience: Teachers (k-12) -- knowledge of BASIC is important for an understanding of this process.

Emphasis: A practical workshop demonstrating the process of merging a LOGO program onto BASIC and then saving for VIDEO.

Objectives:
1. To format a picture diskette with the Apple Tool Kit so that only the LOGO image is saved.
2. To use Koala pad color feature to color the image.
3. To merge the picture diskette image onto a binary file so that the program can be accessed with BASIC.
4. To save the program on a VIDEO recorder and add sound.

LOGO GRAPHICS WORK IN APPLE LOGO, TOO, AND
CAN BE SAVED ON VIDEO FOR DEMONSTRATIONS

by Peggy Hill and Felesa Baker

In March of 1984, my computer partner and I worked on a LOGO program to teach children story patterns. Immediately we had a problem with the computer memory. We questioned the possibility of using LOGO in a BASIC program and were told it couldn't be done. However, that month Hank Voderberg's article (March, 1984) in The Computing Teacher gave us the clue we needed to solve the memory problem. Voderberg had successfully merged a LOGO program onto a BASIC program by first storing it in a binary file. He stated in his article that his program only worked with Terrapin LOGO. In his program, he saved the LOGO picture and merged it into a BASIC program. But he hadn't found a way to do the same thing with Apple LOGO.

We were able to find a procedure which would work with Apple Logo. First, we developed our pictures in LOGO and then saved them onto a PICTURE diskette provided by the Apple Tool Kit. Then to further save memory, and the tedium of recursion for filling in color, we used the Koala Pad. Phillips (1984) article in The Computing Teacher gives another idea for filling in color using Delta Drawing. Finally, with this binary file saved, we were then able to add it to a BASIC program. The directions for this are below.

YOU WILL NEED:

1. Blank diskette
2. LOGO language diskette
3. Apple LOGO Tool Kit Diskette (This is a public domain diskette and most computer stores will allow you to copy the diskette with CopyA of the 3.3 DOS Systems Master.)

BEST COPY AVAILABLE

4. Koala Pad diskette

5. Diskette with LOGO programs already saved or one to which you can write a LOGO program and save image.

6. Two disk drives

PROCEDURE:

1. Boot the Apple LOGO Tool Kit diskette. (#3 above)

   -After the light goes off you will be running in BASIC

   -A] will appear as the prompt.

2. Type:

   NEW (Then press return)

3. Type:

   LOAD PICTURES (Then press return)

4. Remove the Apple LOGO Tool Kit from the disk drive.

5. Insert the blank diskette (#1 above). BE SURE TO DOUBLE CHECK YOUR BLANK DISKETTE. The next procedure will be to save the PICTURES program on another diskette. When you do, you will initialize it and erase anything that is already on that diskette. MAKE SURE IT IS BLANK!

6. Type:

   INIT PICTURES

   This initializes the blank diskette with PICTURES.
   This means that PICTURES will run automatically
   when your diskette is booted.

   (Press Return)

7. Remove this diskette which we will now call the Picture Diskette

BEST COPY AVAILABLE

and boot Apple LOGO in the normal way.

8. Insert your LOGO program diskette (#5 above) when it tells you to.

9. At this point you may create the picture you wish to store or load the one that you will be using.

10. When your picture is loaded or completed and saved on LOGO program diskette remove the LOGO diskette.

11. Insert the Pictures diskette into Disk Drive #1, close the door and type:

.PRINTER 6

(Press Return)

12. This ends the LOGO session and starts the PICTURES program. Your Turtle Graphics drawing will disappear from the screen. However, it is still in memory.

13. Follow the instructions that PICTURES puts on the screen.

type: S (for SAVE)

Then give your picture a name. (Let's set up a phony name. We will call this diskette DIAGRAM. For the rest of the procedures we will refer to this LOGO picture saved onto a PICTURE Diskette as DIAGRAM.

14. This next section will allow you to fill in your LOGO Picture without doing it laboriously in LOGO with recursion. Your will need two disk drives.

15. Boot KOALA Diskette in Drive 1 and place PICTURE Diskette in Drive 2.

16. On the KOALA Menu go to STORAGE option.

a. Select L for LOADING of Disk Drive 2.

b. Backarrow to erase "PICTR" (This is the name which is

in now. You want to erase it and put yours in which is called DIAGRAM.

c. Type: DIAGRAM (or the name you called your LOGO picture and press return.

d. Now return to the KOALA Menu (by pressing black button on tablet).

e. Select FILL and a COLOR.

f. Press the black button and the picture will return. You can fill in with the color you selected by placing your marker in the middle of that area. (Make sure the area is enclosed. Otherwise, the color will "spill" out.

g. When you're finished coloring in this manner, return to the menu.

h. Press STORAGE.

i. Type S for saving picture.

j. Backarrow again as you did in #b. to erase "PICTR."

k. Type DIAGRAM or whatever you have named your file and press Return.

17. What you have to this point is now a LOGO picture with color filled in from the KOALA PAD Diskette and saved onto the PICTURE Diskette.

18. Remove the KOALA PAD diskette from disk drive 1 and PICTURES from disk drive 2..

19. Cold boot PICTURES Diskette in disk drive 1.

Type: Control C (Return)

Type: New (Return)

Write program now in BASIC to call up the picture called DIAGRAM (or whatever you have called it).

20. This is our example below with an explanation for the instructions.

```
]10 PRINT TAB (15) "A SAMPLE PROGRAM" (Title Page Instructions begin here.)
]20 PRINT
]30 PRINT TAB (15) "FOR COMPUTER CLASS"
]40 PRINT
]50 PRINT TAB (20) "by PEGGY HILL"
]60 PRINT TAB (20) "& FELESA BAKER"
]70 FOR I=1 TO 2000: NEXT I          (This is a pause of about 3 seconds.)
]80 HOME
]90 D$-CHR$(4)
]100 PRINT D$;"BLOAD DIAGRAM, A$4000"  (This loads the binary picture file)
]110 POKE -16299,0                   (This is the graphics poke for page 2)
]120 POKE -16302,0                   (Sets graphics to full screen)
]130 POKE -16297,0                   (Sets graphics to high resolution mode)
]140 POKE -16304,0                   (Sets graphics mode)
]150 FOR I= 1 TO 4000: NEXT I        (This is the pause command)
]160 TEXT
]165 PRINT
```

```
1170 PRINT TAB(20) "THE END"
1180 FOR I=1 TO 2000: NEXT !
1190 HOME
1200 END
```

21. Save this program as you do for any BASIC program by Typing SAVE and then the name of the program.

EX: SAVE CALLDIAGRAM

22. Each time the PICTURE Diskette is run now, you will boot and then:

a. Type: Control C (Return)

b. Type: LOAD CALLDIAGRAM (Or whatever you named it.) (Return)

c. Type: HOME (Return)

d. Type: RUN (Return)

There are many reasons for assembling your LOGO program onto BASIC. 1) Because you are only saving the picture, LOGO is loaded onto the computer and displayed much more quickly. 2) Memory is no longer the major problem it was with LOGO because the LOGO language is not taking most of the room in memory. 3) You don't need a LOGO language diskette to access the program. 4) You can chain together many LOGO programs and show them more quickly.

This program was then transferred onto video cassette. In the classroom children were recorded in a narration of their program onto the video cassette. Now we have a computer program which is saved on video cassette with the sound provided by the children. This is also

an excellent process for teachers who have programmed and then want to add voice to their computer program for demonstration purposes, parent's night or whatever.

TO RECORD ON VIDEO

To record on video, plug the computer monitor chord into the video inlet in the video recorder. Then just record as you would a television program. Run your program and record while it is running. For voice addition you will need a video recorder with "dub" capabilities. The process will vary with machines. You will either plug in your microphone, or recorded sound into the audio dub inlet.

Students at the University of Houston University Park enrolled in the course, Computers in the Schools, are also writing LOGO programs, saving them in BASIC, recording them with a video recorder and then narrating them for demonstration purposes. It is then easy for them to take the video cassette into the school and use with other children and/or teachers. Both teachers and students using this process of saving LOGO graphics in the BASIC programs have found this to be a useful tool for making programs for demonstration purposes.

BIBLIOGRAPHY

Phillips, W.R. Filling in LOGO Graphics for the Apple is Easy as Pie! The Computing Teacher, 12, (December/January 1984/1985), p.28.

Voderberg, H. Use LOGO Graphics in Your BASIC Program. The Computing Teacher, 11, (March 1984), p.10-11.

BEST COPY AVAILABLE

STORY GRAMMAR AT THE COMPUTER: MODELS AND PROCESS

by

Peggy Hill

Intended audience: Teachers k-8 (computer background is not necessary)

Emphasis: Models for word processing

Objectives:
1. To structure a story into its plot components.
2. To differentiate between four basic structures and application of one model for writing.
3. To write a story, using a word processing package and SUPERPILOT

STORY GRAMMAR AT THE COMPUTER by Peggy Hill

OBJECTIVES:
1. Participants will be able to structure a story into its plot components.
2. Participants will be able to differentiate between four basic structures and use one model for writing.
3. Participants will be able to write a story, using a word processing package or SUPERPILOT.

RESEARCH

Research in the past ten years has shown that story grammar is an effective structure for teaching reading and writing under certain circumstances. Generally, as a structure for reading comprehension it is only effective under certain instances mainly due to the limited nature of the story grammars, i.e. simple stories with third graders and below. However, as a model for writing, an understanding of basic structures has proven to be a tremendous organizer for children as they begin to write. This workshop will demonstrate several types of story grammars as defined by recent research (Abrahamsom and Shannon, 1983 The Reading Teacher & Sebesta, 1983 The Reading Teacher). These structures are well suited to demonstration and innovations with the computer. Their application using the computer as the tool will be demonstrated with several word processing packages, Kidwriter, Story Maker and Bank Street Story Book along with a demonstration using Super Pilot to initiate the structure. SuperPilot is well suited for an integration of word and pictures.

CONCLUSIONS OF STORY GRAMMAR

The research in story grammar structure has shown the following:

1. The reader uses the structure of story grammar to help him with both comprehension and recall.
2. Young readers especially benefit from direct instruction in story grammar.
3. Knowledge of story grammar helps students to predict outcomes.
4. Knowledge of structure does transfer to other stories resulting in an understanding of more complex structures.
5. Knowledge of story grammar gives a concrete base for writing narrative prose.

PLOT STRUCTURE

A general plot structure is defined as a narrative story which can be structured into characterization, setting, and plot. This will be demonstrated by the following method. Student will fold his paper into six parts and labels each box as follows: who, where, what, action begins, action ends, resolution.

WHO	WHERE	WHAT
Action Begins	Action Ends	Resolution

BEST COPY AVAILABLE

FOUR TYPES OF STORY GRAMMAR

Four of the eleven structures defined by researchers will be presented. These according to Sebesta, Abrahamson and Shannon are the most popular models.

A. CONFRONTATION - Characters meet a problem that must be resolved and most of the action is taken up with solving the problem.

B. QUEST - Characters consciously search for knowledge about themselves and the surrounding world.

C. CONTRAST - Characters have opposing viewpoints, come from contrasting settings, or meet constrasting adventures, or the protagonist engage in a repeated action and get differing points of view.

D. EPISODIC - Episodic stories have incident after incident which reveals the plot by relating the adventures of the characters.

Examples of each will be shown.

Practice:

Participants will then be able to work on one basic plot structure as well as a specific story grammar which has been demonstrated in the workshop.

BANK STREET STORY BOOK

Mindscape, Inc., 3444 Dundee, Northbrook, IL 60062

Grade/Target Age Group: Grades 3 - 9

Subject: Creative Writing and drawing

Hardware Required: Apple II+, IIe or IIc with at least 64K of memory, one disk drive, colored monitor, joystick or KoalaPad

Cost: $39.95

BEST COPY AVAILABLE

SOFTWARE FOR CREATING ILLUSTRATED STORIES

KIDWRITER

Spinnaker Software Corp., 215 First Street, Cambridge, MA 02142

Grade/Target Age Group: Ages 6 - 10

Subject: Creative Writing

Hardware Required: Apple II, II+, IIe or Commodore 64, IBM PC, IBM PCjr
Color monitor recommended; joystick optional; one disk drive

Cost: $29.95

STORY MAKER

Scholastic Inc., 2931 East McCarty St., P.O. Box 7502, Jefferson City, MO
65102

Grade/Target Age Group: 8 years and up

Subject: Creative writing and illustrated text editing

Hardware Required: Apple II+, IIe or IIc (64 RAM minimum), single or dual disk drive, Koala Pad, joystick, or AppleMouse required
Optional printers: Epson (with a Grappler card), ImageWriter or Scribe printer

Cost: $39.95

COMPETITIVE MATH TEACHING STRATEGIES USING THE COMPUTER

Gaye Russell
Alief I.S.D.

This presentation is designed to familiarize the elementary junior high teacher with computer related strategies and activities. Mental shortcuts appropriate for computations without paper and pencil will be demonstrated. University Interscholastic League Number Sense contest format and rules will be discussed.

Math competitions for elementary and junior high students have become common place. Many types of contests, using a variety of formats, are presently being held. Major urban schools currently conduct Math Olympiad Contests within their districts while rural school hold contests incorporating several districts within a geographic region.

The majority of these competitions follow the University Interscholastic League Number Sense format and rules. These rules state that all problems are to be solved mentally. Erasures, markovers, or marks of any kind are not permitted. The test is ten minutes in duration.

Accurate and rapid computation are vital elements to math competition success. Training for speed and accuracy of computation is the goal of both the student and the teacher. To this end, the computer proves to be a valuable tool.

There are several software programs that are conducive to increasing speed and improving accuracy of computations. One program, which is available to most school districts, is the MECC vol. 1 speed drill which allows the student and/or teacher to control the speed with which the problems are displayed. This program also provides individualized reinforcement of basic math

skills, essential to successful competition.

The teaching of mental shortcuts is necessary for successful preparation for competition. Many mental shortcuts that allow the student to compute without paper and pencil are ideally suited for drill and practice on the computer. An example of one shortcut is the FOIL method of multiplying a two digit number by a two digit number. An explanation of this method follows.

The letter F stands for first; which in math refers to the numeral in the ones place. Multiply the digit in the ones place in the first number by the digit in the ones place in the second number.

1) 28 X 36 2) 8 X 6 = 48 3) Record the 8; carry the 4

F 28 X 36 = ___8

The letter O stands for outside, which refers to multiplying the digit in the tens place in the first number by the digit in the ones place in the second number.

1) 28 X 36 2) 2 X 6 = 12

O

The letter I stands for inside, which refers to multiplying the digit in the ones place by the digit in the tens place in the second number.

1) 28 X 36 2) 8 X 3 = 24

I

Add the answers for O and I together as shown below:

1) 12 + 24 = 36 2) Add the 4 you carried. 36 + 4 = 40 3) Record the O and carry the 4.

28 X 38 = 08

The letter L stands for last, which refers to multiplying the digit in the tens place in the first number by the digit in the tens place in the second number.

1) 28 X 36 2) 2 X 3 = 6 3) Add the 4 you carried.

L

6 + 4 = 10
Record the 10.
28 X 36 = 1008

The above is just one example of a mental shortcut. Others will be shared during the presentation. Additional copies may be obtained from the author by writing to her at the following address: 11675 West Bellfort Apt. 1020 Houston, Texas 77099.

SIDE STEPPING THE ESSENTIAL ELEMENTS

by Dena Bankston - Cy-Fair I.S.D.

The students on the middle school level, presently enrolled in Computer Literacy will probably have the chance to at least be exposed to dealings with software uses of word processing, databases, spreadsheets, and computer graphics (Logo). But for at least another two years, and possibly up to four, the students enrolled in your high Computer Math I class may be experiencing their first exposure to a computer.

Taking some "side steps" from the usual curriculum, can supplement these voids in their programming skills, besides adding new life to the momentum of the class structure, and interacting with other disciplines.

The Computer Math I team at Cy-Fair High School in the Cy-Fair I.S.D., implemented two such skills into their curriculum this year, and had such success in doing so that possibly the idea can be expanded for the 85-86 school year.

The population of our classes consists of average or above average Juniors and Seniors. The team felt that Word Processing skills would be helpful, especially to the college bound students. We conferenced with the Jr/Sr English teachers and picked a time frame to teach "Easy Script", in which the students had an English theme due. The paper, when completed would serve as a major grade in both subject areas. The assignment served to develop a new skill to change the pace of the class, and interact with a second discipline (not one usually connected with the math department!).

Teaching computer graphics by the use of Logo, was a second rewarding side step. I particularly chose the week proceeding the Christmas holidays to teach Logo-it offered a pleasant break before Christmas, and the holiday season offered an opportune theme for their projects. An outline was designed to list as few ideas as possible to teach in two days of lecture, yet enough to develop the skills to complete their designs. I felt it important that a history of Logo be included in the introduction. The outline included the topics listed on the attached sheet. During the second day of lecture, one student would type in commands shown as examples so the students could see an illustration of getting around in Logo. A list of commands and important facts were placed beside the computer. The assignment was (either in pairs or individually) to create a holiday design implementing the following:

I. Illustrating the holiday theme
II. Color
III. Use of Repeat Command
IV. Using at least two levels of refinement
V. General appeal and complexity
VI. Run error free from a calling procedure

They had three class periods to complete the assignment. On the fourth day

the projects were loaded on the computers, and graded for completion of the . We added a little holiday fun by inviting selected teachers, administrators, and parents to judge the projects not on programing skills, but on appeal. Each class was awarded a first, second, and third place award.

The students were enthusiastic and eager to work - and all right before a holiday!

Logo Notes

I. Introduction - founder, history, meaning, terms....

II. TURTLE MOVES (IMMEDIATE MODE)

III. SCREEN commands + SYSTEM COMMANDS (WORKSPACE + saving)

IV. DRAWING POLYGONS - CIRCLES

V. T. T. T. T.

VI. REPEAT COMMAND - (nested loops)

VII. VARIABLES

VIII. DEFINING PROCEDURES - (teaching the turtle!) primitives of procedures

IX. MODULAR PROGRAMMING - (TOP DOWN DESIGN)

X. YOUR PROJECTS - REQUIREMENTS

XI. MISC. --- colors, setxy, heading, ...

System Commands

Load "*",8

catalog

Read ^ "filename

Save ^ "filename

Goodbye - erases workspace

POTS - prints out all titles of procedures in workspace

PO ^ [list file names - prints out procedures listed...]

ERASE ^ [list of file names erases procedure from workspace]

Printer Commands

Printer - turns printer on

No Printer - turns printer off

Turtle Commands

FD ^ ___ forward

BK ^ ___ backward

RT ^ ___ right

LT ^ ___ left

PU pen up

PD pen down

HT hide turtle

ST show turtle

PC ^ ___ sets pen color

BG ^ ___ sets background color

HOME - sends turtle home

SET XY ___ ___

HEADING

PC-1 pen erase

Screen Commands

CS - clears screen

DRAW - clears screen & sends turtle home

<f1> text screen

<f3> split screen

<f5> full screen

Edit Commands

CNTL C - defines procedure

CNTL G - aborts run

CNTL D - deletes character

CNTL O - opens line

CNTL K - kills rest of line

CNTL N - sends cursor to next line

CNTL P - sends cursor to previous line

CNTL A - sends cursor to beginning o line

CURSOR KEYS !!!

Colors

0 - black	8 - orange
1 - white	9 - brown
2 - red	10 - light red
3 - cyan	11 - gray 1
4 - purple	12 - gray 2
5 - green	13 - light green
6 - blue	14 - light blue
7 - yellow	15 - gray 3

BEST COPY AVAILABLE

COMPUTER LITERACY FOR THE SMALL COMMUNITY
Design & Implementation of an Adult-Based Course

Pam Wood
P.O. Box 669
Anahuac, Texas 77514

With present emphasis on computers in our society and in public education, many small communities are finding themselves in need of continuing education classes. Successful implementation of such a class depends upon several factors. Consider first the need for such a course. Is there an availability of computer-education resources? Is the community close to junior colleges and/or major universities that offer similar courses through continuing education programs? Even if the answers to these questions are "yes", there still might be a place for a locally sponsored course. Many people are intimidated by "going to a college" yet will feel very comfortable in a home-town class environment. So geographic location should be considered in determining need.

Another factor is the support of the school district administration. Many districts will view this course as an added burden and responsibility, and will totally discourage the idea. Other districts will consider it a community service and support the program 100%. Public relations, therefore, within the district is important for successful implementation of such a program.

The commitment of one or more people is necessary to see this program through the planning stage. The teacher assigned to teach the course should be involved in deciding times, course content and class size. The commitment of this teacher is important as the course will require night classes and many teachers would oppose being given an assignment such as this.

The actual planning stage involves specific decisions about instructor, class size, dates and salary so that a proposal can be presented to the school board. Keep in mind that many decisions will depend upon school district rules and policies. Some recommendations, however, will be suggested. Ideally, the class should be taught in a computer lab, especially if the course is going to be offered during the school year. It would be difficult to transport 10-20 computers, set them up and move them back in one evening. For summer classes, this would not be as difficult because the computers could remain in the specified room during the course. This decision, more than any other, will be dependent on present set-ups within a school

district. Some specific recommendations can be made as to class size, salary and dates. Class size should be limited to a maximum of 20 adults with a minimum of 10 computers. The reasons for maximum class sizes are basically the same for adults as for children. Teacher/pupil ratio should be kept as low as possible for better instruction. There should never be more than two adults to a computer and preferably one adult to each computer. If the class is at the maximum limit of 20, it is suggested the instructor have a large TV monitor hooked up to a computer at the front of the room. This will help the students know where they should be and will be a good "teacher's aide".

In a recent survey of schools offering computer classes, it was found that salaries and tuition varied. One district responded that they set their 15-hour course class size at a maximum of 12 and paid the instructor all of the proceeds. The tuition was $30 per person. In this case, the school district took in no revenue from the course. A second school district offered a 25-hour adult class allowing for a maximum of 20 registrants and paid their instructor approximately $18-20/contact hour. The tuition for this course was $75-100 with half-price breaks for school district personnel. Another district charged $50 tuition for a 18-hour course. The instructor was paid $40 per person and the district kept $10 per person to cover some expenses. Clearly, salaries and tuition vary among districts.

When considering dates and times to offer a class, it is recommended that no more than two sessions per week be taught for a period of four to six weeks. A minimum of 18 hours is needed to adequately cover the areas of computer literacy. The dates should not be set for Monday through Thursday nights, or some similar arrangement of consecutive nights. Adults are usually busy and involved in the evenings and it is very difficult to commit for three hours per night, four to five nights in a row. Remember, too, that most adults have worked all day. A break about half-way through the session, with coffee/cold drinks and light snacks, would serve to relax and alleviate any anxieties. The instructor should be available during this 15-minute break to discuss or answer any questions that might have arisen during the first half of the evening.

With these decisions made a proposal can be presented to the school board. Gaining school board approval can be accomplished several ways. One is to first get acceptance/approval from the curriculum director. After this one might need to personally go to the superintendent for approval and finally approval from the school board. School districts have different procedures but at some point it must

BEST COPY AVAILABLE

be decided who makes the final proposal to the board ... the teacher, curriculum director, or the superintendent.

If approval is obtained, the public must be informed. Newspaper releases and public service announcements on TV or radio would be the most efficient means of notifying the public. A good news release will be one that discusss computer literacy and briefly outlines the course content. Some method of pre-registration should be offered. A number might be given to call and sign up or a tear-out, mail-in form could be printed in the newspaper.

Actual course content has not been addressed as it is assumed the main essential elements outlined by TEA will be considered. However, most emphasis should be on applications (word processing, data base, and spread sheets) in which adults are most interested. History and societal impications might be discussed a little each session and one session might be given to some simple BASIC programming. A suggestion for the last session might be to invite representatives of local computer stores to come and speak for 20 minutes each and exhibit their education/home-use software for preview. This session might also be open to the public.

After school board approval, public awareness and course content finalized, the implementation phase of the adult course is complete. Now it is time to sit back and enjoy the class. The school district, the instructor and the community will all benefit from an informative and exciting course such as this.

TRIALS AND TRIBULATIONS OF COMPUTER LITERACY

Rita Dupree

Computer Literacy is a new course that is being required by the state of Texas. Our purpose today is to share together some of the ideas and resources for teaching the course next year. Included here is an example of the handouts.

WORD PROCESSING/HISTORY

I. Introduction of Word Processing
   - A. Overheads from word processing section of state materials
   - B. Demonstrate word processor and commands
   - C. Allow students to enter information into word processor

II. History
   - A. Filmstrip (EyeGate the History of Computers)
   - B. Overheads from history section of state material
   - C. Allow students to enter notes into word processor
   - D. Computer History Disk I (Britannica Series
   - E. Assign report to each student on some phase of history (use material in history section of state material as introduction)
   - F. Puzzles from state materials
   - G. Handout A Bit of Computer History from Walch

III. Using the Word Processor
   - A. Use move command to arrange notes in some order
   - B. Use replace to correct spelling errors
   - C. Use erase to take out words not needed
   - D. Use insert to make sentences complete
   - E. Enter, correct and print out assigned report on word processor

WORD PROCESSING

A TOOL FOR LEARNING

By: Martha Lawrence

As educators we have many resources available for our use. One that should be of great value is a word processor.

Word processors can be of much help to the teacher. It can make paper work much easier and faster to do. Much of the paper work we do must be modified as the years pass. Student records are up dated; tests are generated and filed; and inventories can be maintained.

All this paper work cuts into preparation and teaching time. The word processor cannot eliminate the paper work, but it can decrease the time spent entering, correcting and modifying it.

What can a word processor do? It can do anything a typewriter can do and more. With a word processor a paper can be typed in, corrections made, paragraphs re-arranged, inserted or deleted or replaced entirely with the pressing of

a few keys. Once a rough draft has been entered, it can be quickly modified and stored on a floppy disk. Then it can be easily retrieved, revised and printed again and again. Because of the ease with which a word processor can be used, writing will lend itself to greater creativity and originality. In addition to the ease of use, dot matrix printers make excellent ditto masters. Print-outs can also be used to make transparencies in a matter of minutes.

In addition to the uses by the teacher, word processors can be used as a teaching tool in many ways with the students. Students who do not do well in writing assignments can become efficient writers. Many of these children do not do well because of mechanical errors. It takes too long to make changes, or even just to make it readable. Now, they have a tool to use to make the mechanics easier. It no longer takes hours to edit and correct their papers. They can now use this time to refine and sharpen their thoughts and ideas. The use of the computer encourages the student to revise and edit their written work. It havs been determined

through classroom research that students using the word processor, write longer papers and edit their work more closely. Revising becomes just that and not just busy work.

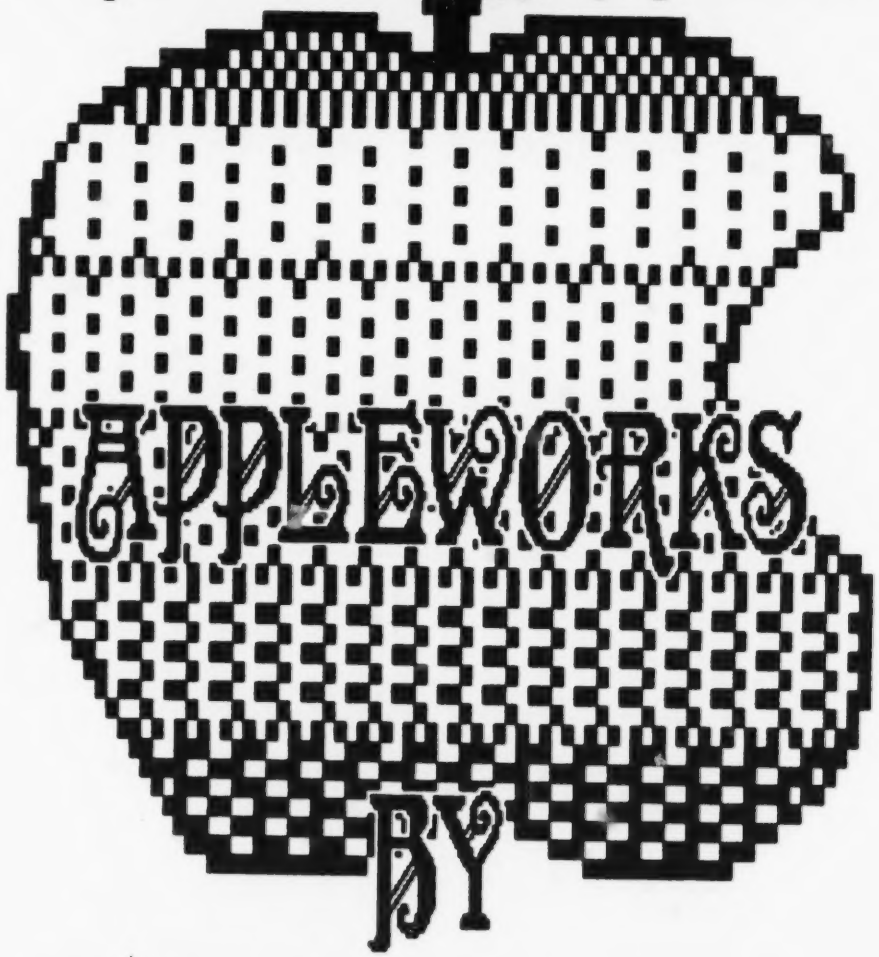
As we have discussed some of the benefits of using word processors, we must also look at some barricades in using the word processor. One of the most often voiced disadvantages is the limited access to computers. It takes a tremendous amount of time to compose, edit and revise their work. One way to work around this problem is to use it as a group writing project, several students working together to produce a finished project.

Another stumbling block can be poor key boarding skills. I do not consider this a major problem because corrections are so each to make. In addition to this, it will not be too long before the computers we use are voice synthesized. Commands will be entered verbally.

Another handicap to overcome is faculty resistance. The best way to over-come this is to have them use it. After some hands on activities even the most skeptical will be more willing to try it. They will soon discover how much time they can save when producing stencils, personalized letters to parents.

When using word processing programs, the important thing to remember is that it is a tool for learning.

WELCOME TO THE BIG APPLE!



PAMELA CHO

Welcome to the Big Apple! Appleworks

Contents

Introduction
1.1 Appleworks Defined
1.2 Equipment Needed
1.3 Copying Disks and Backing Up Files
1.4 Desktop and Clipboard Concept--How Appleworks Works
1.5 Special Features
1.6 Getting a File
1.7 Creating a New File
1.8 How to Cut and Paste

Word Processing

2.1 Appleworks Word Processing
2.2 Useful Commands
2.3 The Professional Touch
2.4 Ideas to Try

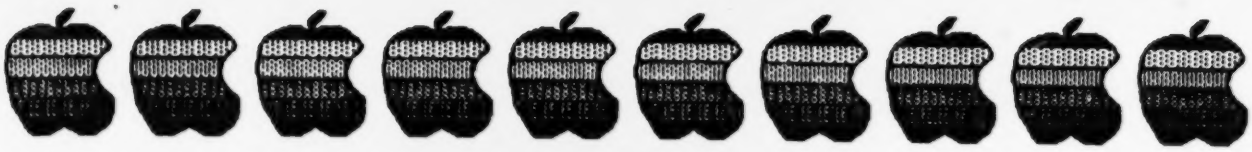
Data Base

3.1 Appleworks Data Base
3.2 How to Create a Data Base
3.3 Useful Commands
3.4 Ideas to Try

Spreadsheet

4.1 Appleworks Spreadsheet
4.2 How to Create a Spreadsheet
4.3 Useful Commands
4.4 Formulas with Appleworks Spreadsheet
4.5 Formulas for Calculating
4.6 Ideas to Try

Conclusion
Resources



Presenter: Ms. Pamela Cho
District: Galveston Independent School District
Position: Computer Literacy teacher, Eighth Grade Center

INTRODUCTION

The purpose of this workshop is to give participants a hands-on experience with an integrated program called Appleworks. In the eighth grade Computer Literacy program in Galveston, students learn to use the three-part program. The Computer Literacy curriculum requires application software, and I feel Appleworks is an excellent choice.

1.1 Appleworks Defined:

Appleworks is a computer program that allows you to do data base management, word processing, and spreadsheet analysis. Because Appleworks is an integrated program, you can move back and forth among the three applications, and combine the three kinds of information onto a single document.

1.2 Equipment Needed:

Appleworks users need the Appleworks Startup and Program disk. The training disk, tutorial manual, and sample files disk are helpful in learning to use the program. The reference manual is essential.

If you have an Apple II e, you need an 80-column text card, preferably with extended memory, along with a monitor, and one or two disk drives. Apple II c users need a monitor. A printer makes maximum use of Appleworks.

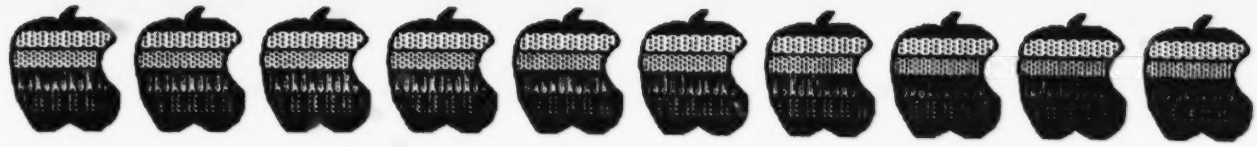
1.3 Copying Disks and Backing Up Files:

The Appleworks manual gives permission for you to back up the program for your own use with any standard copy program, such as COPYA on the system master. To make a data disk that would store files, use Applework's OTHER ACTIVITIES menu option 5, FORMAT A BLANK DISK, to format your disks.

1.4 Desktop and Clipboard Concept--How Appleworks Works:

The files you create or put into memory from your data disk, go to the "Desktop", which is really computer memory (RAM). You can actually have 12 files on the Desktop, but you can examine only one at a time. When you're finished with them, you can take them off the Desktop and save them permanently on a disk.

The clipboard temporarily holds part of a file that you've cut from one place and want to paste in another. For example, you can cut a student's grades from the spreadsheet, print it to the clipboard, and paste them into a letter to the parent in the word processor.



1.5 Special Features:

One of the best features of Appleworks is that the commands that you use are simple to remember. Appleworks makes good use of mnemonics with its "open-apple commands." For example, if you wish to save your file, hold down the open-apple, and press S. To move a sentence, open-apple and M. In other words, the commands are easy to remember, and most can be used in each application.

Appleworks also makes use of friendly warnings that help prevent accidents. Unfortunately, Appleworks has an ever-present "Control G" beep that occurs when you do something the program isn't expecting.

1.6 Getting a File:

Choose ADD FILES TO THE DESKTOP.
Choose THE CURRENT DISK.
Replace the Appleworks program with the data disk, and press return.
The files on the disk will be listed on the screen. Move the cursor using the up or down arrow to the file you want, and press return.
Take out the data disk and put back the Appleworks program.

1.7 Creating a New File:

Choose ADD FILES TO THE DESKTOP.
Choose either 3, 4, or 5 for word processing, data base, or spreadsheet.
Choose FROM SCRATCH.
Type a name for the new file, press return.

1.8 How to Cut and Paste:

If you want to paste either the data base or spreadsheet onto the word processing, it's easy. Let's say you want to paste your spreadsheet grades to your parent letter. First, create the letter and the spreadsheet. In the spreadsheet, press open apple P for Print. Choose ALL, then choose THE CLIPBOARD. Next get to your letter by pressing Open apple Q(uick). Put your cursor where you want the grades. Press Open apple M(ove), and choose PASTE FROM CLIPBOARD. Voila!

Word Processing

2.1 Appleworks Word Processing:

The Appleworks Word Processor lets you compose text and revise memos, letters, reports, proposals, and articles...whatever you want, and painlessly edit it, rearrange it, polish it up with various options, and print it.

2.2 Useful Commands:

Delete key - erases one letter to the left
Open apple C(opy) - duplicate text
Open apple E(dit) - switch from "insert cursor" to "strikeover cursor"
Open apple D(elete) - delete text
Open apple F(ind) - locate text
Open apple H(ard copy) - print screen "dump"
Open apple K(alculate) - calculate pages
Open apple N(ame) - change file name
Open apple M(ove) - move text
Open apple O(ptions) - printer options
Open apple P(rint) - print document
Open apple Q(uick Change) - go to a file on the Desktop
Open apple S(ave) - save file
Open apple Y - clears line to the right of the cursor
Open apple Z(oom) - examine chosen printer options
Open apple arrows - move around document
Open apple 1 - 9 - Appleworks "ruler", open apple 1 goes to top of document, open apple 5 goes to the middle.

2.3 The Professional Touch:

Appleworks has some very nice printer options. While working with the word processing document, you may wish to center a title, underline it, justify the report, use boldface, super- or subscript, double space, etc.

First position your cursor before the text that you, for example, wish to center. Press open apple O(ptions). Displayed are the list of options that you may use. Type the two letter code, CN, and press return. Press ESC to leave Options.

2.4 Ideas to Try:

Most students are required to turn in reports to their English, Science, and Social Studies teachers. Why not encourage the use of word processing for these reports? Students practice computer skills, putting them to practical use. I encourage students to use graphics programs, such as Printshop, by Broderbund, to design cute covers for their reports.

Letter writing is important. Students can write applications for jobs, complaint letters, fan mail, pen pals in other computer classes, thank you notes, invitations, etc.

Students also can type in notes or homework using the word processor. Neat work is important, and many students prefer the word processor and the printer for doing class assignments.

Data Base

3.1 Appleworks Data Base:

The Appleworks Data Base allows you to work with information you normally keep in lists: names and addresses, schedules, inventories.

The data base consists of the file name, category names, and records (information under the category names.)

3.2 How to Create a Data Base:

Follow the guidelines for creating a new file in 1.7, choosing Data Base as the option. After typing the file name, you are to type in the category names. First, delete "Category 1" by pressing open apple Y. Then, type in the category names, such as STUDENT ADDRESS, pressing return after each label. After the last category, press return. Press ESC to continue. Then Appleworks will allow you to type in the information by the category labels, one record at a time.

Always press return when you finish typing or correcting an entry. Pressing ESC after typing an entry will restore the former entry.

The following is an example of an Appleworks data base:

Restaurant	Price	Food	Rating
Creative Eats	moderate	American	Good
Sahara Fires	expensive	Arabian	Spicy
Ocean Blue	cheap	Sea food	Fair
Bistro	expensive	French	Great
El Papa	moderate	Mexican	Good
Red Dragon	moderate	Chinese	Terrific

Data Bases can be printed out in table format, as in the above data base, or in label format, as in evelope labels, for mailing purposes.

3.3 Useful Commands:

The commands in the word processor are the same for the data base with a few additions or changes.

Open apple A(rrange) - arrange category alphabetically, numerically, or chronilogically
Open apple D(elete) - delete category or record
Open apple I(nsert) - insert new category or record
Open apple J(ustify) - left justify a category in a label format
Open apple L(ayout) - arrange, widen, shrink categories in desired layout
Open apple R(ecord Selection) - select out records. Example: Science report on Deserts could show a list of all deserts located near mountains in the Northern Hemishere.
Open apple T(otal) - Total a numeric category
Open apple Z(oom) - Switch between single- and multiple-record layout
TAB key, open apple TAB - move from column to column

3.4 Ideas to Try:

Data bases that students can develop are friend's names, addresses, phone numbers, hobbies. Also, students are quick to make data bases of favorite music groups, and song titles.

Other ideas would be to use data bases as a reference tool, such as Spanish vocabulary words with English translations, hard-to-pronounce words with phonetic spellings and definitions, Earth Science mineral charts, famous Americans and achievements, computer software reviews, library book title, author, and subject. Any kind of inventory would be appropriate, computer equipment and cost, for example.

Spreadsheet

4.1 Appleworks Spreadsheet:

The Appleworks Spreadsheet lets you create information you' normally put on a spreadsheet and manipulate with a calculator: budgets, financial forecasts, income and expense statements, gradebooks.

4.2 How to Create a Spreadsheet:

After following the procedure for Creating a New File, 1.7, choosing spreadsheet, you type in the spreadsheet information, labels, values, and formulas.

Spreadsheets are organized by cells, much like the grids on a map. You refer to cells by the letters (column) and numbers (rows), such as cell G4 or cell B2. Cells that have non-numeric information are "label cells", and cells that have number values or formulas are "value cells".

The following are examples of spreadsheets:

Barry Riptide's Computer Budget

Item	Cost
Apple II e	$800.00
monitor	$200.00
disk drive	$200.00
Appleworks	$275.00
Total:	$1,475.00

Grades for Barry Riptide

English		Science	
quiz	90	test	88
hwork	80	lab	99
test	70	hwork	72
class	99	test	100
hwork	88	lab	92
AVG:	85.4	AVG:	90.2

4.3 Useful Commands:

The commands are the same for the spreadsheet as the word processor, except for the following.

Open apple B(lank) - blank out the contents of a cell
Open apple J(ump) - jump to other window (see Open apple W)
Open apple L(ayout) - change values to percents, dollars, etc., change size of categories
Open apple U - change contents of a cell
Open apple W(indow) - make two windows, or one window, so that you can "freeze" the labels, and still view all parts of the spreadsheet.

4.4 Formulas with Appleworks Spreadsheet:

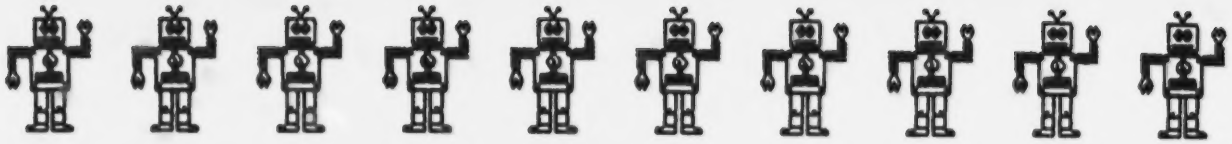
Formulas are what make spreadsheets so invaluable. They can be compared to using variables in math. The variables are the cells, the values in those cells are used by the formulas to produce a number.

For example, a student types the names of two classes, English in cell A1, and Science in cell C1.

Next, the grades are put in, English grades in cells A3 through A15, and Science grades in cells C3 through C15.

At the bottom of the list of grades in cell A20 the formula for calculating the average for English would be the formula: @AVG(A3.A15). Similarly, Science averaging formula would be @AVG(C3.C15), meaning "average the values in C3 through C15."

** Note ** When typing values, the numbers are automatically right-justified. Labels are automatically left-justified. Layout arrangement can be done after typing in the information.



4.5 Formulas for Calculating:

The following is a list of formulas that can be used. I think that spreadsheets are easier to learn if you use one type of formula per spreadsheet, such as totaling (summing up categories), or averaging (such as averaging grades.) Students and teachers are much less frustrated!

For averaging:

@sum(a1.a20)	Average cells A1 through A20

For adding, subtracting, multiplying, dividing:

@sum(a1.a20)	Add cells A1 through A20
@sum(a1+a20)	Add cells A1 plus A20 (just two cells)
@sum(a1-a20)	Subtract A20 from A1
@sum(a1*a20)	Multiply A1 times A20
@sum(a1/a20)	Divide A1 by A20

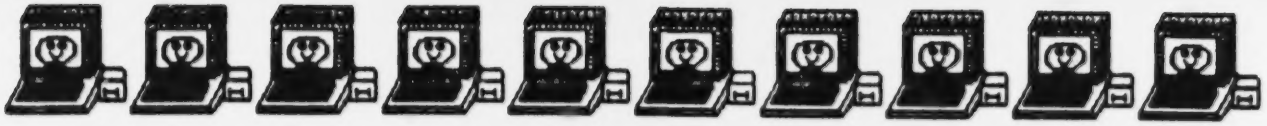
There are many more types of formulas available with Appleworks, including square root, integer, ordinal positions, absolute value, and others.

4.6 Ideas to Try:

When teaching the spreadsheet, begin with cursor movement. Then show students how to identify cells by letter and number. Next proceed to typing a label and a column of values in specified cells. Then type in a formula to average or sum the values in the list. Next go back to the values, typing in a different value. The average or sum at the formula cell automatically changes.

After students become accustomed to the typing of labels and values, have them make a complete gradebook for their classes. As they receive new grades, they put them in, their averages automatically changing.

Other ideas are sports percentages, catalog orders, software prices, personal expenses, hobby costs, computer expenses and comparisons, retail store inventories, and personal assets.



Conclusion

Appleworks is a very complete program which I have used with my students very successfully. Because Appleworks is so diverse, students of all academic levels find it useful. I use it to produce nearly every document possible, including this handout, worksheets, tests, the department inventory, budget, letters, gradebook, and others. Appleworks now has a spelling program sold separately that would help correct typing errors. The name is Sensible Speller for ProDos, produced by Sensible Software, which retails for $125.00. Appleworks, made by Apple Computer, retails for $250.00.

The greatest part about this program is that once the students have become accustomed to the program commands in word processing, then the data base, and spreadsheet are not difficult, since the commands are mostly the same.

Also, the "cut and paste idea" is fantastic, since separate word processing, data base, and spreadsheet programs can mean printing out separate pages, cutting, taping, and "xeroxing" them on one sheet! Appleworks eliminates tedius retyping and xeroxing.

Teaching Appleworks takes approximately six weeks. Spending at least two weeks per application is essential for students to be able to become accustomed to and make use of the program.

My students and I have enjoyed using Appleworks. If you have Apple computers, I highly recommend this software. Many other word processing programs exist for the Apple, PFS, Bank Street, Word Juggler, but I feel that this program is more useful because of the combined data base and spreadsheet.

Resources:

1. Appleworks, Apple Computer
2. Printshop, Broderbund
3. Sensible Speller for ProDos, Sensible Software
4. Follett Library Catalog, Follett Quality Courseware
5. Appleworks Reference Manual, Apple Computer
6. Appleworks Tutorial, Apple Computer
7. Quick Reference Card, Apple Computer

UNDERSTANDING DATA BASE MANAGEMENT

Dr. Terry A. Olivier
Emporia State University

This paper is designed to provide a general understanding of data bases and how they are organized. The presentation will be a "hands on" approach to designing and using a public domain data base management program. Ideas for using data bases within the classroom setting will be discussed.

A data base is an organized collection of information. Schools use data bases to keep track of student records, employee information and administrative costs. Using a data base program is like using an electronic filing cabinet or a computerized index card box. Data base management software is flexible in that it allows the user to create a data base and then categorize, store, retrieve, and update information quickly and easily.

Common characteristics of data bases include: a) user definition of parameters, b) creation of reports and lists that are organized based upon user defined criteria, c) allowance for easy update and inquiry into files, and d) most are general in nature as opposed to specific applications.

There are two types of data bases. The first type is the relational data base. This type of program uses the data itself to mesh logically related files. Data can be gathered on any basis the user desires and the program allows the user to go directly to the desired information without intermediate search steps. Relational data bases offer users the ability to compare information in one file with information in another file by "relating" the two (or more) bits of information

through one common field. This type od data base is generally more powerful and considerably faster. DBASE III is an example of a relational data base program.

The second type of data base program is hierachical in nature. There are several levels of data storage in the hierachical data base each more detailed than the previous level. For the user to reach a given piece of data the program must search through successive levels of the data. This can be time consuming if the data base is very large. PFS:File and Perfect Filer are examples of hierarchical data base programs. It is this second type of data management program that is generally found in the schools.

School files can contain a great deal of information. For example, each teacher may want to maintain such specific facts about her/his students as address, telephone number, age, and grade point average. Each one of these items is a field. Therefore, the name of a student would be a field as would the student's address.

A collection of all the fields that relate to one student constitutes the student's record. A record then is a set of information about one subject in the file. A collection of all the related records is a file. The school file would contain the records of all the students within a particular class.

The data base program has many instructional applications in the classroom. Teachers as well as students will find data base programs very useful. Students can create data bases in almost every subject. These data bases can be used to explore

relationships, classify, and sort data. Teachers on the other hand will find that they can keep track of their students and organize their teaching materials using a data base management program.

Copies of a handout entitled "Ideas for Using Data Bases in Your Classrooms" can be obtained by writing to the author at the following address: Dr. Terry A. Olivier, Emporia State University, 1200 Commercial St., Box 36, Emporia, Kansas, 66801.

GRAPHIC SOFTWARE: WHICH ONES TO PURCHASE AND HOW TO USE THEM

Gerald Pollard
Texas School for the Deaf
1102 South Congress
Austin, Texas 78764

This presentation looks at the various software packages that enable students and staff to create interesting and useable graphics. The Koala Pad, Blazing Paddles, Pixit, Flying Colors, Apple Mechanic, Zoom Graphix, Image Printer II, The Professional Sign Maker, and Print Shop will be demonstrated and evaluated.

More and more graphic utilities are coming on the market and are making it easier for students and staff to develop and utilize meaningful graphics. I'll try to limit this information to a few of my favorites and describe them briefly . First I'll describe the software that is primarily for producing graphics, then I'll describe the screen dump software, and then two programs that create graphics and also dump them and then a few comments about slide projection programs.

PRODUCTION One of the easiest to use utilities is the Koala Pad. This device plugs into the game paddle socket and let you draw using a pencil like object much like you would write/draw on paper. The screen menu assists in selection of the various drawing aids such as fill, ray, box, circle, erase, color, draw, lines, and magnify. Preschoolers as well as adults enjoy using this device. One disadvantage is that no text can be typed onto the graphic created. To solve this problem you can use Blazing Paddles, Pixit, Flying Colors, or Apple Mechanics. Each of these packages allows the retrieval of a standard binary hi-resolution graphic file so that text with different fonts and colors can be added to the pictures. An effective combination is the Koala Pad and Blazing Paddles software in that (1) you then have the advantage of adding text in several type styles, (2) adding shapes from the shape libraries, (3) direct screen dump capabilities,(4) mixing up to 200 different colors and (5) the micro feature which allows easy editing control over each pixel (smallest dot in a hi-resolution graphic) while simultaneously viewing the total picture. This feature saves a great deal of frustration when only the blow-up

is available as in the case of the Koala Pad software.

Pixit is another nice program that allows retrieval of an existing graphics and to type labels or captions on them and also it has a shape editor which allows the creation of different pictures. In addition, stored on the disk is 128 pre-drawn shapes. Pixit also lets you dump your picture to a Epson printer with a Graftrax interface card. Both Pixit and Blazing Paddles contain the same fonts and shape tables. Unless you need a shape table editor purchase only the Blazing Paddles software. It's more versatile.

An additional software package that allows text on top of graphics is Flying Colors. Flying Colors is similar to the Koala Pad menu with all the box, circle, brush, line, and micro aids available. This package also creates a slide presentation and more about that later.

A fourth package is Apple Mechanics. It is unique in that various files from this disk can be loaded onto your personal disk for use.The xtyper file allows you to load three different fonts out of a choice of six on this disk and with the Typeface disk there are 26 font styles from which to choose. Once loaded the text can be placed on top of any graphic.

SCREEN-DUMP PROGRAMS Once you have created your graphic masterpiece you will need to send it to a printer to show the world your beautiful work. Pixit, Flying Colors and Blazing Paddles will dump graphics to the printer but not like Zoom Graphix or Image Printer II. Zoom Graphix can print any hi-resolution graphic vertically, horizontally, inverse, or in a positive or negative format. It can zoom in for only a portion of the picture or enlarge the picture approximately four times its original size. It supports 21 printers and 20 interface cards, is menu-driven and, is easy to use.

Image Printer II is another screen dump program which works similarly to the Zoom Grpahix but has three additional features. First, it is not copy protected and can be loaded onto your personal disk for screen dump capabilities. Secondly, one can capture pictures from other disks (even copy protected disks) and then edit or modify them prior to dumping them to the printer. A third feature allow you to shrinks an existing picture making it one half its original size prior to printing. Zoom Graphix and Image Printer II are both equally easy to use. If purchasing only one, get Image Printer II because it does more and can be loaded on a program disk.

SIGN MAKERS If you are in need of large, black and white lettered signs or charts, then the Professional Sign Maker is for you. This program allows you to create bold signs with letters one to six inches tall. Also there is control of border selection, text size, format, variable spacing between letters, and how dark or light the printing will be. One special feature is that you can save up to 40 different signs on this disk for printing out later. This program does require you to read if not study the manual prior to use. However, once the concepts are understood, operating the program is relatively simple.

The Print Shop software package was the best-seller during the recent Christmas season for several dealers because it is easy to use and it does so much. You can create personalized greeting or thank-you cards, invitation and announcements, letterheads, large banners (much better than MECC's), signs, and more. It comes with 60 pictures and designs from which to select and with an additional disk you can get 125 more pictures.The font selection contains a choice of 8 different styles varying in size. The greeting cards, letter heads, and banners made with Print Shop are of high quality. Also include is a graphic editor which allows you to draw your own graphics with the keyboard, joystick, or Koala Pad. This package will become a standard in the industry because of its ease of use. Four limitations come to mind. Only graphics created with Print Shop are useable, there is no final viewing of the created graphic prior to printing, it takes about four minutes (which seems slow) to print out a greeting card and images created cannot be saved to the disk.

SLIDE PROJECTORS Occasionally, one might use several graphic pictures to make a presentation. Flying Colors provides the best slide program. One is asked to select in order which hi-resolution graphics they want to show. Maximum capacity is 12 for Flying Colors. One can either show each graphic manually (with forward or back control) or have then shown automatically.

PRODUCT INFORMATION FOR APPLE II

APPLE MECHANICS
Beagle Bros Micro Software Inc.
4315 Sierra Vista
San Diago, California 92103
1-800-227-3800 ext. 1607
List Price: Unprotected disk, $29.95 (Typeface disk, $20.00)

BLAZING PADDLES

Baudville
1001 Medical Park Drive, S.E.
Grand Rapides, Michigan 49506
List Price: Protected disk, $49.95; backup disk, $5.00

FLYING COLORS
Computer Color Works
3030 Bridgeway
Sausalito, California 94965
415-331-3022
List Price: Protected disk, #39.95

IMAGE PRINTER II
Sensible Software, Inc
6619 Perham Drive
West Bloomfield, Michigan 48033
313-399-8877
List Price: Unprotected disk with manual, #49.95

KOALA PAD
Koala Technologies Corp.
3100 Patrick Henry Drive
santa Clara, California 95052-8100
408-986-8866
List Price: $125.00

PIXIT
Baudville
1001 Medical Park Drive,S.E.
Grand Rapides, Michigan 49506
List Price: Unprotected disk, $49.95

PRINT SHOP
17 Paul Drive
San Rafael, California 94903
415-479-1170
List Price: $49.95

THE PROFESSIONAL SIGN MAKER
Sunburst Communications
Room Q8
39 Washington Avenue
Pleasantville, New York 10570
1-800-431-1934
List Price: Protected, one disk, backup, teacher's guide $59.00

ZOOM GRAPHIX

American Eagle Software, Inc
P.O. 46080
Lincolnwood, Illinois 60646
312-792-1227
List Price: Protected disk, $49.95

INTRODUCING PASCAL THROUGH KAREL THE ROBOT

by Dell Simon, assistant professor
Department of Computer Science
University of St. Thomas
3812 Montrose Blvd.
Houston, Texas 77006

This presentation will demonstrate a simple programming language which can be used to introduce students to the concepts of structured programming and the syntax of the Pascal programming language, without variables and data structures. First, the world in which Karel lives and his capabilities will be introduced. Secondly, the programming language which controls Karel in his world will be demonstrated. The advantages of using Karel to introduce Pascal will be discussed, followed by suggestions and cautions for implementation.

I. What is Karel the Robot?

Karel the Robot (Pattis, 1981) is both a robot on a computer screen and the language that controls the robot's actions. Just as the computer screen version of a Logo turtle looks very little like a real turtle, the Karel robot looks very little like a real robot. Karel is a simple graphic entity whose environment or "world" consists of the intersection of two half-planes. This world is bounded on the south and west, and crisscrossed with streets and avenues at regular intervals.

The language that controls the robot's actions consists of a well defined set of instructions which may be used to direct Karel to move about his "world" and to accomplish prescribed tasks. For example, Karel may be directed to leave his bed, retrieve his newspaper, and return to bed. He may also be directed to run a steeplechase in which he jumps hurdles of unspecified height and width.

Karel's world consists of a grid of streets and avenues which he can traverse. In his world, he may encounter walls which have been erected. Karel must navigate around the wall sections, which may form hurdles, mountains, rooms, or mazes. In addition to Karel and wall sections, the world may contain beepers that Karel is directed to pick up, carry, or put down.

II. How is Karel programmed?

The primitive instructions for directing Karel are:

1) move — Karel moves one block forward

2) turnleft — Karel turns 90 degrees to the left

3) pickbeeper — Karel puts a beeper in his beeper bag

4) putbeeper — Karel places a beeper on the corner

5) turnoff — Karel turns himself off

Any of these instructions may be combined to form new instructions by defining new instructions. For example, a turnright instruction may be defined by the following command:

```
DEFINE-NEW-INSTRUCTION turnright AS
  BEGIN
   turnleft;
   turnleft;
   turnleft
  END;
```

After this definition has been added to Karel's list of instructions, it may be invoked by name, turnright. The concept of "defining new instructions" is similar to Logo's concept of defining a new procedure and thereby increasing one's vocabulary. In Pascal, this concept becomes writing a procedure or function.

The language includes conditional instructions of IF/THEN and IF/THEN/ELSE, and two repetition instructions of ITERATE and WHILE/DO. These four instructions operate on a test of the following conditions which may occur in Karel's world:

1) front-is-clear, front-is-blocked
2) left-is-clear, left-is-blocked
3) right-is-clear, right-is-blocked
4) next-to-a-beeper, not-next-to-a-beeper
5) facing-north (or south or east or west),
   not-facing-north
6) any-beepers-in-beeper-bag
7) no-beepers-in-beeper-bag

These statements may be combined to define new instructions. For example, the following statement demonstrates the WHILE/DO instruction used to remove all beepers from a corner.

```
DEFINE-NEW-INSTRUCTION clear-corner-of-beepers AS
  BEGIN
   WHILE next-to-a-beeper DO
     pickbeeper
  END
```

Most importantly, the language incorporates the concept of block structuring found in Pascal and also exactly the same rules for separating statements with semicolons. This is truly a gentle introduction to BEGIN/END pairs and semicolons.

III. Why teach Karel?

Teaching Karel the Robot as a prelude to Pascal allows students to begin programming without the calculation and attending "math anxiety" experienced when programming tasks emphasize variables and computation. One experienced classroom teacher expressed the opinion that in programming Karel, she had finally found a computer programming language that did not require that one be "good at math". The tasks for Karel are spatial tasks. Solving those tasks is similar to solving a spatially oriented game or puzzle. Not all students will enjoy these kinds of tasks, either.

Top down design can be practiced right from the beginning with Karel, and procedures (DEFINE-NEW-INSTRUCTION) are found to be immediately desirable. The use of semicolons, begin/end pairs, if/then/else statements, and the while/do loop are an integral part of programming Karel.

Karel can be taught as the first two weeks of a high school programming class or perhaps, as part of a junior high computer literacy class. Karel is a natural second language for Logo students who have learned to manipulate the Logo turtle.

IV. Suggestions and cautions

The text for Karel the Robot is extremely well done, with challenging tasks, a multitude of examples, and humor. Karel can be studied strictly utilizing the text, or a combination of the text with the Karel the Robot software, and perhaps with the courseware disks which include all examples and assignments from the text.

Karel is available in versions for several micro amd mainframe computers. The particular version with which I have personal experience is written for the Apple 2e/2+. Students may write their Karel programs using the Apple Pascal editor and filer, and then submit them to the Karel software for compilation and execution. This allows students to become familiar with the Apple Pascal editor and filer, before tackling the Pascal language. My students found the method rather frustrating since we had twenty one class members and only one set of Karel software. Teachers

planning to use this version of the software will need to be creative in classroom management techniques. Small class size, additional sets of Karel and Apple Pascal software, and extra lab time will help to alleviate the situation. Ample hardware also needs to be available. Apple 2+ computers need a language card for both Apple Pascal and Karel software. One should be extremely cautious about attempting to use either Apple Pascal or Karel software without double disks drives for each computer. The educational benefits of Karel can quickly be negated by frustration with lack of hardware, software, and time.

Reference

Pattis, R. E. (1981) Karel the Robot, A Gentle Introduction to the Art of Programming with Pascal. New York: John Wiley & Sons.

BEST COPY AVAILABLE

USING LOGO TO PROGRAM KAREL

By

Nancy Siragusa

St. Michael School

In my first encounter with Karel the Robot, I realized that the Logo turtle could be programmed in much the same way. Of course the Logo turtle does not have "eyes" and cannot detect walls or beepers in its path. This prevents use of conditional instructions such as "if front is clear" or "while right is blocked". However, the simple programming assignments in chapters 1,2, and 3 can easily be transferred because they contain none of these conditions.

There are several advantages in using Logo to perform Karel's tasks. First, the programs can be written without the expense of the Pascal or Karel software. Students can see if their ideas will work by putting the turtle through Karel's paces. Second, I feel that any time students convert programs from one programming language to another, their learning in both languages is strengthened.

In setting up the turtle to perform Karel's work, I designed the primitive instructions first. These include move, turnleft, putbeeper, pickbeeper, origin, and grid. Of course origin and grid are not instructions, but they needed to be defined so that the turtle, like Karel, will have a pre-determined starting position. It also needs a world of streets and avenues.

The instruction "move" is defined as PU FD 10. This will allow the turtle to cover 10 spaces without leaving a trail. "Turnleft" is defined as LT 90, as all of Karel's turns are 90 degrees. In defining putbeeper, I instructed the turtle to draw a

small rectangle (1 x 2) and return to its starting position. This makes a spot that is large enough to be seen over the lines of the grid. Pickbeeper is just the opposite, assuming that the rectangle is already on the grid, the turtle will erase it.

The grid is designed with 10 x 10 squares, and the procedure origin sets up the turtle at the corner of 1st St. and 1st Ave. facing east.

Once these primitive instructions are defined, it is very easy to adapt the simple Karel programs. In the program P36, which I have used as an example, Karel plants a field of beepers to spell out the message "HELLO". After designing this program using Karel's language, it was very simple to write it for Logo. The procedures were essentially the same. In fact, I did this program in Logo first and was able to debug it before running it on the Karel software. The only errors that are hard to detect are the syntactic ones, as semicolons and indentations are not used in Logo.

The Logo language is a good way to introduce Karel and Pascal programming because many students are already familiar with Logo and the way it is structured. If necessary, the primitive commands can be buried in the Startup Aids so the students would not have to load them in every time. Once the students become used to using Karel's commands in Logo, it will be easier for them to use the Karel software.

The following pages include:

1. Logo primitive command sheet.
2. Logo program P36
3. Karel program P36
4. Karel's world with the turtle at the poirt of origin.
5. Program P36 drawn on the grid.
6. Program P36 without the grid.

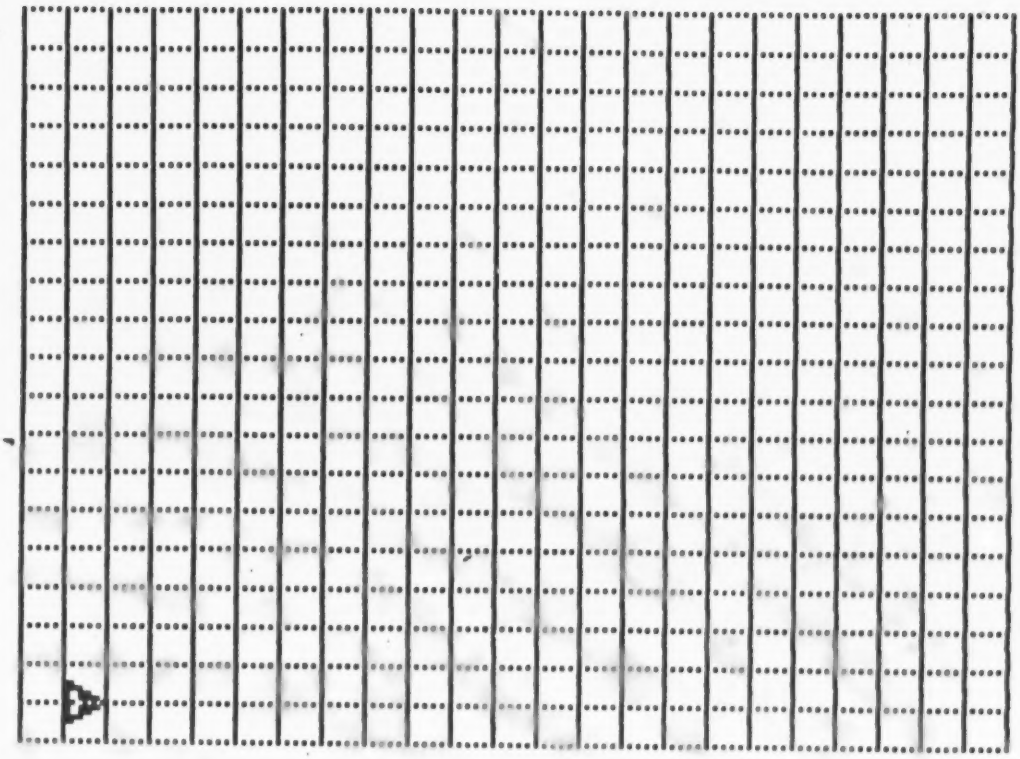
```
?PO [MOVE TURNLEFT PUTBEEPER PICKBEEPER GRID LINES ORIGIN]
TO MOVE
PU
FD 10
END

TO TURNLEFT
LT 90
END

TO PUTBEEPER
PD BK 1
REPEAT 2 [FD 2 RT 90 FD 1 RT 90]
FD 1 PU
END

TO PICKBEEPER
BK 1 PE
REPEAT 2 [FD 2 RT 90 FD 1 RT 90]
PU FD 1
END

TO GRID
HT
PU SETPOS [-110 -70]
PD REPEAT 12 [FD 190 RT 90 PU FD 10 RT 90 PD FD 190 LT 90 PU FD 10 PD LT 90]
LINES ORIGIN
ST
END

TO LINES
LT 90 PU FD 10
PD REPEAT 9 [FD 230 RT 90 FD 10 RT 90 FD 230 LT 90 FD 10 LT 90]
FD 230 RT 90 FD 10 RT 90 FD 230
END

TO ORIGIN
PU
SETPOS [-100 -60]
SETH 90
END

?
```

```
PO (PLANTROW PLANT.2 TURNRIGHT PLANT.H PLANT.E)
TO PLANTROW
PUTBEEPER
MOVE
PUTBEEPER
MOVE
PUTBEEPER
MOVE
PUTBEEPER
MOVE
PUTBEEPER
END

TO PLANT.2
PUTBEEPER
MOVE
PUTBEEPER
END

TO TURNRIGHT
TURNLEFT
TURNLEFT
TURNLEFT
END

TO PLANT.H
PLANTROW
TURNRIGHT
MOVE
TURNRIGHT
MOVE
MOVE
TURNLEFT
PLANT.2
TURNRIGHT
MOVE
MOVE
TURNLEFT
MOVE
TURNLEFT
PLANTROW
END

TO PLANT.E
PLANTROW
TURNLEFT
MOVE
PLANT.2
TURNLEFT
MOVE
MOVE
TURNLEFT
PLANT.2
TURNRIGHT
MOVE
MOVE
TURNRIGHT
PLANT.2
END

?
```

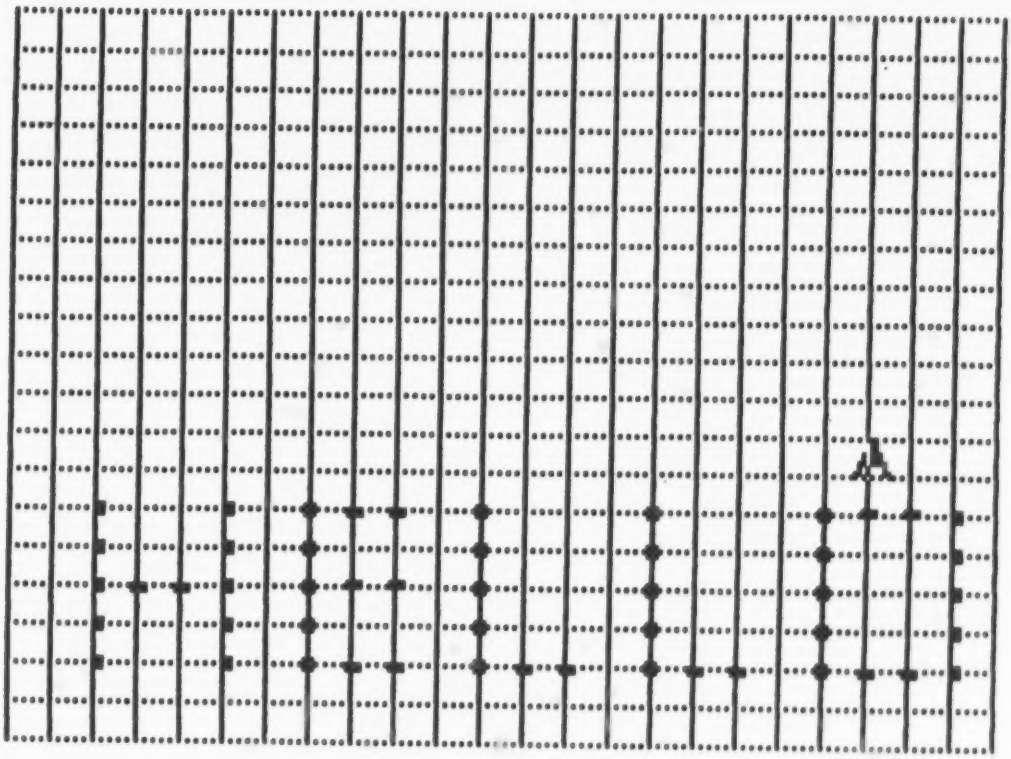
```
PO (PLANT.L PLANT.O MOVE.FROM.L P36)
TO PLANT.L
PLANTROW
TURNLEFT
MOVE
PLANT.2
END

TO PLANT.O
PLANTROW
TURNLEFT
MOVE
PLANT.2
MOVE
TURNLEFT
PLANTROW
TURNLEFT
MOVE
PLANT.2
END

TO MOVE.FROM.L
TURNLEFT
MOVE
MOVE
MOVE
MOVE
TURNRIGHT
MOVE
MOVE
TURNRIGHT
END

TO P36
ORIGIN
MOVE
TURNLEFT
MOVE
PLANT.H
TURNRIGHT
MOVE
MOVE
TURNRIGHT
PLANT.E
MOVE
MOVE
TURNRIGHT
PLANT.L
MOVE.FROM.L
PLANT.L
MOVE.FROM.L
PLANT.O
TURNRIGHT
MOVE
END

?
```

```
BEGINNING-OF-PROGRAM

  DEFINE-NEW-INSTRUCTION plant-row AS
  BEGIN
    putbeeper;
    move;
    putbeeper;
    move;
    putbeeper;
    move;
    putbeeper;
    move;
    putbeeper
  END;

  DEFINE-NEW-INSTRUCTION plant-2 AS
  BEGIN
    putbeeper;
    move;
    putbeeper
  END;

  DEFINE-NEW-INSTRUCTION turnright AS
  BEGIN
    turnleft;
    turnleft;
    turnleft
  END;

  DEFINE-NEW-INSTRUCTION plant-h AS
  BEGIN
    plant-row;
    turnright;
    move;
    turnright;
    move;
    move;
    turnleft;
    plant-2;
    turnright;
    move;
    move;
    turnleft;
    plant-row
  END;

  DEFINE-NEW-INSTRUCTION plant-e AS
  BEGIN
    plant-row;
    turnleft;
    move;
    plant-2;
    turnleft;
    move;
    move;
    turnleft;
    plant-2;
    turnright;
    move;
    move;
    turnright;
    plant-2
  END;

  DEFINE-NEW-INSTRUCTION plant-l AS
  BEGIN
    plant-row;
    turnleft;
    move;
    plant-2
  END;

  DEFINE-NEW-INSTRUCTION plant-o AS
  BEGIN
    plant-row;
    turnleft;
    move;
    plant-2;
    move;
    turnleft;
    plant-row;
    turnleft;
    move;
    plant-2
  END;

  DEFINE-NEW-INSTRUCTION move-from-1 AS
  BEGIN
    turnleft;
    move;
    move;
    move;
    move;
    turnright;
    move;
    move;
    turnright
  END;

  BEGINNING-OF-EXECUTION
    MOVE;
    TURNLEFT;
    MOVE;
    move;
    plant-h;
    turnright;
    move;
    move;
    turnright;
    plant-e;
    move;
    move;
    turnright;
    plant-l;
    move-from-1;
    plant-l;
    move-from-1;
    plant-o;
    turnright;
    move;
    turnoff
  END-OF-EXECUTION
END-OF-PROGRAM
```

BEST COPY AVAILABLE



'78



73

MANIPULATING THE CHARACTER GENERATOR

"How do you get text on the screen with your pictures?" is the first question asked by programmers new to high-resolution graphics. After manipulating the text window in the low-resolution screen, students are eager to combine text and graphics, but often they find the options unsatisfactory.

Commercial utilities offer a variety of both fonts and problems. While characters are varied, integrating the commercial program with the student created one often strains the capabilities of young programmers. Further complications arise because many school systems cannot afford, or will not invest in, graphics utilities. Those that do often find themselves burdened with copyright restrictions that severely limit use.

Public domain software, such as the hi-res character generator, does allow the programmer easy access, but unfortunately the standard fonts are often hard to read on color monitors. The attached utility allows programmers to create original fonts for use with the hi-res character generator. These new fonts can be drawn on grids that are six spaces wide and eight spaces tall and then transferred to a new binary file. This file will replace the character table accompanying the character generator and can be smoothly integrated into programs using the old table.

To use this utility, BLOAD the character table you want to change, BLOAD HI-RES CHARACTER GENERATOR, and then load the program CREATE CHARACTERS.

A grid appears on the screen showing the character designated as CHR$(0). The user is given the option of changing the character (Y), leaving it as is (N), or exiting the program (F). If the character is to be altered, then the first line of the eight line grid appears inversed with the cursor flashing at the first changeable spot. New characters are drawn with the 1 (number 1) and the space bar keys. The 1 key creates an impression or outline, while the space bar moves the cursor one space to the right. Corrections, or backspacing, are done with the left arrow key. Press the RETURN key to move to line two or to leave a blank line. When all eight lines have been filled in, the new character will appear on a hi-resolution screen. Press any key to return to the program. Either accept this new character by pressing the space bar or redo it by pressing the R key. Continue this process through all 127 characters. Provisions for saving the new table are included in the program.

The ability to create new characters gives student programmers sophisticated flexibility. They can use codes, hieroglyphics, or even alphabets from other languages for their own programs. Characters also animate easily and provide excellent subroutines for game programs.

Betty Carter
Barbara Edwards
Ken Ismert

```
1  REM  *** CREATE CHARACTERS ***
3  FOR I = 1 TO 7: READ BV(I): NEXT
4  DATA  2,4,8,16,32,64,64
5 LOMEM = 24576:AD = 26624: GOTO 300
7  HTAB 1: VTAB J + 1: INVERSE : PRINT  SPC( 8): NORMAL
     : HTAB 1: VTAB J + 1
10 B = B / 2: IF B< ' = 0 THEN  RETURN
15  IF  INT (B) = B THEN  INVERSE : PRINT " ";: NORMAL
     : GOTO 10
20 B =  INT (B): PRINT " ";: GOTO 10
50  HOME : PRINT "CHR$(";K;")  AD ";AD: FOR J = 1 TO 8
     :B =  PEEK (AD): GOSUB 7:AD = AD + 1: NEXT J: RETURN

100 C = 1:H = 21:V = 1 + M:L = 6: VTAB V: HTAB 19: PRINT
     M;" ";: INVERSE : PRINT  SPC( 8): NORMAL : VTAB V
     : HTAB H
110  VTAB V: HTAB H + C: IF C > L THEN 190
120  GET A$:D =  ASC (A$): IF D = 13 THEN 210
130  IF D = 8 THEN 160
140  IF D = 32 THEN  INVERSE : PRINT " ": NORMAL :C =
     C + 1: GOTO 110
150  IF D = 49 THEN  PRINT " ":B = B + BV(C):C = C + 1
     : GOTO 110
155  GOTO 120
160 C = C - 1: IF B >  = BV(C) THEN B = B - BV(C)
170  INVERSE : PRINT " ": NORMAL : IF C <  = 0 THEN C =
     1
180  GOTO 110
190  GET Y$:D =  ASC (Y$): IF D = 8 THEN 160
200  IF D < > 13 THEN 190
210  VTAB V: HTAB 32: PRINT B:X =  FRE (0): RETURN
300  FOR K = 0 TO 127: GOSUB 50
310  VTAB 1: HTAB 22: PRINT "CHANGE Y/N/F ?";: GET A$:
      IF A$ = "N" THEN 500
315  IF A$ = "F" THEN 510
320  IF A$ < > "Y" THEN 310
330 AD = AD - 8: FOR M = 1 TO 8:B = 0: GOSUB 100: POKE
     AD,B:AD = AD + 1: NEXT M
345  HGR : POKE 54,0: POKE 55,96: VTAB 12: HTAB 20: PRINT
      CHR$ (K): GET A$: POKE 54,189: POKE 55,158: TEXT

350  VTAB 12: HTAB 21: PRINT "OK R/⟨SP⟩ ? ";: GET A$: IF
     A$ = "R" THEN  FOR N = 2 TO 9: VTAB N: HTAB 19: PRINT
      SPC( 17): NEXT N: GOTO 330
500 X =  FRE (0): NEXT K
510  HOME : PRINT "DO YOU WISH TO SAVE THIS ?";: GET A
     $: IF A$ = "N" THEN 550
520  IF A$ < > "Y" THEN 510
530  PRINT : INPUT "ENTER NAME OF FILE : ";N$
540  PRINT  CHR$ (4);"BSAVE ";N$;", A26624, L1024"
550  HOME : PRINT "DO YOU WISH TO SEE THE CHARACTER SE
     T ?": GET A$: IF A$ = "N" THEN 600
560  IF A$ < > "Y" THEN 550
570  HGR : POKE 54,00: POKE 55,96
580  FOR L = 0 TO 11: PRINT " "; CHR$ (L);: NEXT L: FOR
     M = 13 TO 127: PRINT " "; CHR$ (M);: NEXT M: GET
     A$
590  POKE 54,189: POKE 55,158: TEXT : HOME
600  PRINT : PRINT "FINISHED"
700  END
```

MANIPULATING MEMORY ON THE APPLE II COMPUTER

Jeffrey W. Bloom
College of Education
University of Houston - University Park
Houston, TX 77004

The experience of computer programming can be very rewarding. The reward usually comes after hours of attempting to figure out why this creation of ours does not work. Syntax error after syntax error is followed by weird results. However, when all the bugs are worked out and the program works flawlessly, a sense of great accomplishment and pride ensues. Confidence arises in the programmer, who then embarks on more complicated and grandiose projects. Adding high resolution graphics and more detailed interaction are typical enhancements amateur programmers try to include. With such changes the problems encountered in those first programs seem like child's play. The scenario follows the pattern of writing a good program without any of the graphics. This part is tested and all the bugs are removed. Then high resolution graphics are added. Anxiously, the programmer starts the program. All is working well, then out of the great electronic void...nothing happens. The program stops. The programmer's pulse increases, a cold sweat breaks out. The program is LISTed. Streams of "garbage" scroll up the screen..."I didn't write that! Where did that come from?!! Everything was working fine!" Hopefully, this programmer saved the program before trying to run it. What appeared as garbage on the screen was indeed garbage in the computer's memory. The same thing can happen with very long programs, usually simulations, that contain a large number of strings. This type of program may not stop running for quite some time, but the end result is the same.

These two types of errors involve the way the Apple computer uses memory. In computers with 48K and above Random Access Memory, there is approximately 36K of memory available. In other words, the programmer has room for about 36,000 characters of program, strings, variables, and graphics. That is 36K bytes of memory.

When a program is loaded into memory it is normally put at the bottom, which is at 2048. If a program is 4,000 bytes long, it ends at 6048. What is called "LOMEM" is set at this point by the Apple. All numeric variables are stored above this position as the program runs. If 300 bytes of variables occur during the execution of this program, free memory starts above 6348K. String variables, on the other hand, build down from the top of memory, or from HIMEM, which is located

at about 38K. String variables, unlike numeric variables, do not reuse the same memory location. So that as string variables are re-used, they are stored at different locations. The effect is that as the program runs, the strings start building downward. If 100 bytes of strings are initially used in our sample program, then memory is free from 37,900 down to 6348. If the strings are reused 15 times in the course of the typical use of this program, free memory starts downward from 36,500. This sample program is small and no problems would occur. However, if the program is 30,000 bytes long (about 115 sectors on the disk) LOMEM would be set at 32048. If 500 bytes of strings are used initially, after 12 repeated uses of these strings, the strings would start to "eat" the program. The memory locations of the program at 32K would be replaced by the strings.

The solution for this problem is fairly simple. Apple has a house cleaning statement that will clear the old string memory locations: FR = FRE(0). Placing this statement at various locations in the program will prevent strings from gobbling up programs.

The use of high resolution graphics complicates this situation. There are two "pages" of hi-res memory. These "pages" are areas of memory, each of which is 8K long (actually 8,192 bytes). The clincher is that these pages start at location 8,192 (8K) and extend to 24,576 (24K). Using page 1 (8,192 to 16,384), which is determined by the command HGR, leaves enough room for 6,144 bytes of program and variables. If a program that uses page 1 is 7,000 bytes long, when HGR is executed 1000 bytes of the program are destroyed along with all of the numeric variables and arrays. One alternative is to use page 2 (HGR2), which starts at 16,384 and ends at 24,576. That adds 8K of room, but you do not have the option of a text window at the bottom of the screen (at least, it is not easily available).

The other possibility is to rearrange the way in which memory is used. Instead of having your program load at the bottom, under page 1, you can load your program above page 1. That provides a total of about 22K of memory for program, variables, and strings. If both pages of hi-res graphics are to be used, then the program could be loaded above page 2. To do this a "driver program" can be used to reset the new locations and RUN the program. This is what it looks like:

TO LOAD ABOVE PAGE 1:

```
10 REM *** DRIVER FOR PROGRAM ***
20 D$ = CHR$(4)
30 POKE 103,1 : REM *** START AT 1 BYTE ABOVE 16384
40 POKE 104,64 : REM *** 64 X 256 = 16384
```

```
50 POKE 16384,0 : REM *** PUT A 0 (ZERO) JUST
   IN FRONT OF WHERE PROGRAM WILL LOAD ***
60 PRINT D$; "RUN PROGRAM"
```

TO LOAD ABOVE PAGE 2:

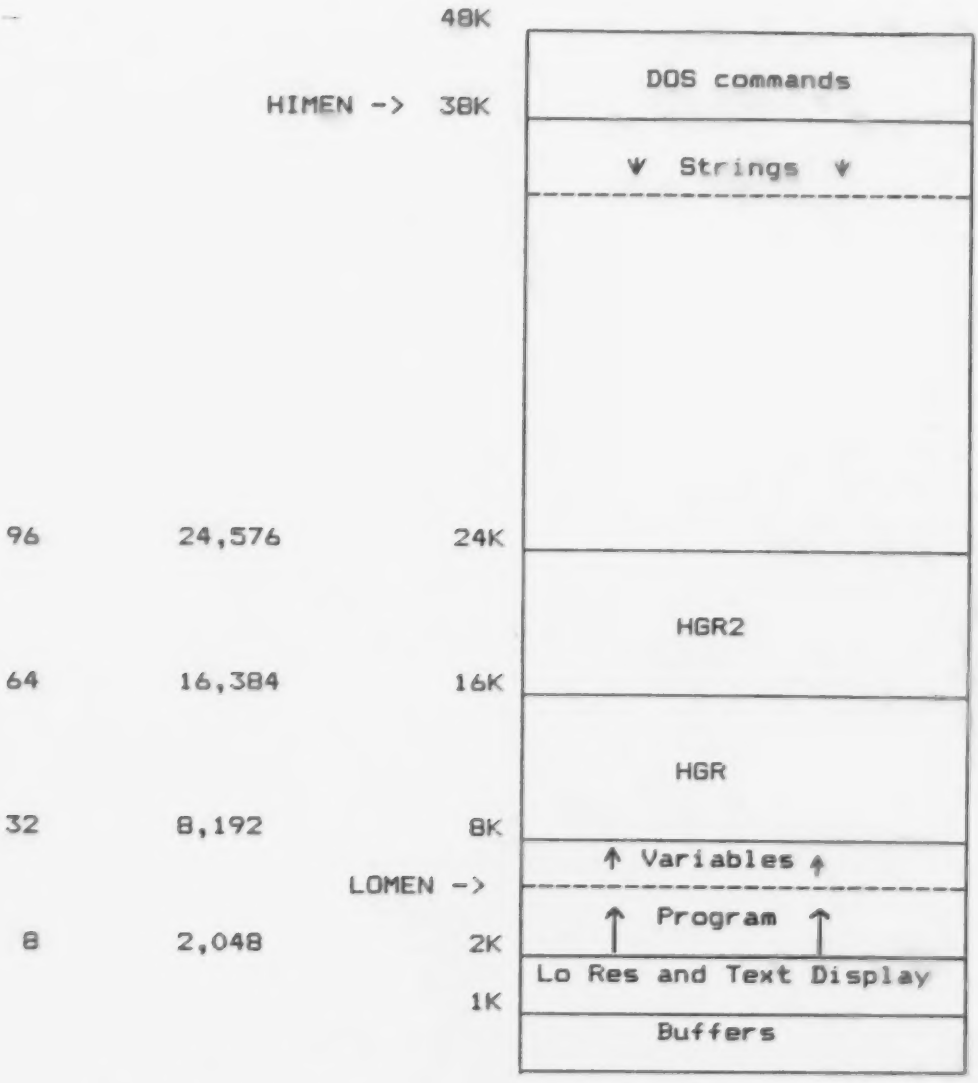
```
10 REM *** DRIVER PROGRAM ***
20 D$ = CHR$(4)
30 POKE 103,1
40 POKE 104,96
50 POKE 24576,0
60 PRINT D$, "RUN PROGRAM"
```

That is all there is to it! But, you must remember to reset the loading locations back to normal in your program's END routine. If you do not do this, when you attempt to run or load another program you will get an error message and the program will not run. This routine could look like this:

```
1000 REM *** END ROUTINE ***
1010 POKE 103,1
1020 POKE 104,8
1030 POKE 2048,0
1040 PRINT "END"
1050 END
```

These simple commands should be a start towards making life much easier when confronted with more complex programming tasks. The memory map on the next page serves as a handy reference for figuring out where your program is and how to deal with potential problems. To figure out how long your program is: (1) turn on the computer with a disk that does not automatically load a program in drive 1, (2)type: FR = FRE(0):PRINT FR, (3) record the number, (4) load your program, (5) repeat steps 2 and 3, and (6) subtract the result of step 5 from the result of step 2 and you have the number of bytes in your program. The program could be run and then the same procedures repeated to determine the amount of space used by variables and strings. From that point it is just a matter of figuring out how you will arrange the use of memory.

APPLE II MEMORY MAP



INTERACTIVE VIDEO AND GRAPHIC SPECIAL EFFECTS

G E O M E T R Y

Lois E. Gordon Moseley

GOAL

It is the intent of this production to bring about an awareness of shape and the realization that shape is an attribute by which objects differ and by which they can be classified.

RATIONALE

The current trend in geometry is in the direction of informal and intuitive learning experiences. These help build associations between real world experiences and more formal abstract geometric concepts.

INSTRUCTIONAL DESIGN

An introductory audio visual exploration of the environment gives way to extended vocabulary and formal classifications in self contained sections.

Testing, remediation and reinforcement is as follows: 1) Correct responses to test items located at the conclusion of each section results in a positive visual reinforcer. (2) Incorrect responses will receive an audio remediation followed by a second opportunity to respond to the test item. (3) Two incorrect responses to the same test item produces the test item with the correct answer.

Student and computer share in the control and the pace of the program. Students select the content area and the computer selects the content, manner of presentation and appropriate reinforcement or remediation.

OBJECTIVE

Students will be able to identify, define and list attributes of geometric shapes.

EFFECTIVE USE OF COMPUTERS IN THE SCIENCE CLASSROOM

Jeffrey W. Bloom
College of Education
University of Houston - University Park
Houston, TX 77004

The rapid introduction of computers in schools has prompted many researchers and educators to criticize school districts for jumping on the "bandwagon" of the new technology without proper planning. Without adequate forethought and research, we as educators are in danger of failing to use microcomputers effectively. Student outcomes may not justify the continued expenditures on hardware and software. In effect, the fate of microcomputers could follow the fate of previous curricular movements, such as, the open classroom and Biological Science Curriculum Study (BSCS). Lack of research and effective planning and training can have disasterous effects on the future of technology in education.

Fortunately, the pace of research has been increasing. Early results are pointing the way to a successful future, but we as the implementers will carry the responsibility for the successful use of computers. In this regard, the primary rule of thumb should be one of maintaining a critical, yet inquisitive attitude toward their use. We should strive to keep up with the research literature, at least in the form of summaries of the research. In the classroom, student responses should be monitored and informal studies performed. In other words, certain key questions can be kept in mind as we continue and expand the use of computers in the classroom. Are students motivated by this package? Are students learning what they should be learning? Are students using their time more efficiently? These are just examples, however teachers should be generating questions that are relevant to their specific needs.

Uses of microcomputers in the classroom and laboratory

There are two primary categories of computer use (1) computer managed instruction (CMI) and (2) computer based instruction (CBI). Computer managed instruction involves uses that make a teacher's job easier. The following list provides an easy reference to some of these possibilities.

Managing student records
- attendence
- grades
- addresses and phone numbers
- textbook records
- anecdotal records

Managing information	- resource list (people) - resource list (suppliers) - electronic bulletin boards - mainframe databases
Managing instruction	- individualized instruction progress reports - group investigation progress reports - mastery learning management - testing (formative and summative)
Accounting	- equipment inventories - supply inventories - departmental budget
Communications	- form letters and reports to parents - forms and hand-outs - lesson plans and notes - memos - general writing
Other	- graphing and graphics - statistical analysis

For the most part, existing software can handle most of these tasks. The initial time spent in setting up data bases to keep records, will save time later. Data bases, form letters, and other formats can be set up before school starts. Then, while in the midst of teaching, only a minimum amount of time is needed to keep records current, send reports to parents, run off assignments, add up and average grades, and so on. In addition, you might need information about some topic. By calling into a major data base (The SOURCE, Knowledge Index) or other source, the information needed can be received within minutes.

Computer based instruction, on the other hand, involves student use of the computer. The following list provides an overview of potential uses.

Remediation	- drill and practice - tutorials - learning games
Demonstration	- tutorials - simulations (of events) - simulations (of experiments)

Problem Solving	- problem solving games
	- simulations (interactive)
	  - performing experiments
	  - manipulating variables in systems
	  - making decisions in simulated real life situations

Research	- data collection - data analysis - information gathering (from data bases) - communicating with other classes doing similar research - programmming to meet specific needs - record keeping - calculating
Production	- graphing results - using a word processor to write the final report

Although the list may seem a bit overwhelming, the idea is to start out slow and add activities. Software should be evaluated before it is used. Each package or other computer use should be analyzed in terms of its potential impact on learning effectiveness. A couple of guidelines to consider when looking at software are (1) that it should not replace laboratory experiments or good teaching and (2) that it should present a content area or process that is difficult to present by any other means. Software and other computer uses should facilitate learning.

A new product that is being marketed by HRM Software allows various measurement instruments (temperature, force, light intensity, and sound analysis) to be connected to a computer. In conjunction with their software, students can perform experiments while the data is being simultaneously displayed on the screen in graphic form. The instructional result is that students (1) are able to manipulate more variables in less time, (2) can become more actively involved in the scientific method (analyzing and questioning the results as they proceed through the experiment), (3) can maintain their level of interest (by eliminating lengthy calculations), and (4) can obtain much more accurate results than they would by using hand-held instruments.

Communication devices (modems) can allow students to communicate with other science classes around the country (or world). Potentially, students could arrange to do similar experiments where geographic location could be a critical independent variable. Both classes could perform the same experiment and then share their results.

Modems can, also, allow students who working on research projects to tap powerful sources of information, such as, large data bases, bulletin boards, and other larger computers.

Data base and word processing programs provide students with the opportunity to keep organized records and to formulate high quality finished products. The drudgery and frustration of making changes and corrections in records and papers is considerably reduced. When the product is complete, a sense of pride can result.

The potential uses are extensive. The imagination and skillfulness of the instructor are the essential ingredients.

Summary of the research

Briefly, the patterns emerging from the research literature indicate that certain types of software are effective. Tutorials and drill and practice are most effective in science and foreign languages and with low and high achievers. Students tend to learn more in less time, but retention is questionable. Affective outcomes, such as, attendance, motivation, attention span, and independence, are affected positively.

Unfortunately, research is lacking in the area of problem solving, simulations, and other uses of computers. However, teachers can pursue a more active role in research. Informal studies in each class help the individual teacher, but collaborative efforts with university researchers could benefit vast numbers of teachers and education as a whole.

Concluding remarks

In order to successfully put computers to their best use in the science classroom and laboratory, careful planning and evaluating are of utmost importance. On the other hand, fear of trying something new will not yield new and exciting results. Computers are powerful tools, that if used properly, can transform a passive science curriculum into an active and exciting process of discovery. Even remediation can become more rewarding for both the teacher and the student.

Project SERAPHIM

Computer Workshop

NSF-Science Education

EASTERN MICHIGAN UNIVERSITY

Department of Chemistry Ypsilanti, MI 48197 U.S.A.

HANDS ON IN THE SCIENCE CLASSROOM

- Nan Porter -

There are various ways to use the microcomputer in the science classroom. These methods will be presented briefly. Each piece of computer software best fits into only one or two of these classroom settings. Software types and classroom settings will be discussed. Evaluation forms that address classroom settings are included in this packet.

Lab interface kits will be demonstrated. These kits were built from materials readily available through Radio Shack or a similar electronics store. Very little money was required to build the kits which connect into the game port of an Apple computer.

Participants will review and evaluate computer software, most of which is available through Project Seraphim for $4.00 per disk. Project Seraphim currently has over 100 programs on 26 disks. Since the project does not allow copying of software and the $4.00 fee is designed to encourage honest purchase of materials, no copying of software will be done at this workshop. Order forms are included in this packet.

Presenter is Dr. Nan Porter from Kinkaid School. Address any additional questions to her at 201 Kinkaid Drive, Houston, Tx. 77024.

SELECTING SERAPHIM SOFTWARE FOR CHEMISTRY

Where does one begin with the SERAPHIM software which is now available for the Apple II? This has become a formidable problem now that there are over 100 programs distributed over 26 disks. Below are the recommendations of James George of DePauw University. These are most suitable for high school introductory and second year courses.

Class Demonstrations

1. Bohr Atom (Rittenhyouse); disk 201
2. Hydrogen (Ragle); disk 201
3. Quantum Mechanics (Groves); disk 202
4. VSEPR (gelder); disk 301
5. Titration Curves (Meyer); disk 501
6. Reaction Rates (Moskaluk); disk 601
7. Molecular Speed Distribution (Blanck); disk 603
8. Decay (Groves); disk 1001

Out-of-Class assignments

1. Rutherford's Experiment (Ground); disk 202
2. Formula (Bendall); disk 301
3. Boyle (Bendall); disk 401
4. Charles (Bendall); disk 401
5. Rast 2 (Bendall); disk 501
6. Rates (Bell); disk 601
7. Faraday 2 (Bendall); disk 603
8. Lake Study (Whisnant) disk 804

Drills

1. Electron Arrangement (Arlotto); disk 202
2. Balanced Equations (Masterman); disk 302
3. Naming (Masterman); disk 303
4. Mole Calculations (Groves); disk 304
5. Balloon (Groves); disk 402
6. Molarity (Dees); disk 502
7. pH (Dees); disk 502
8. Solubility (Susskind and Pascual); disk 502

Educational Games

1. Dalton (Bendall and Powell); disk 403
2. Chemical Dungeons (Estell); disk 602
3. Waqual (Estell); disk 802

Project SERAPHIM/CCCE Evaluation Form

Name of Software _____ Author/Publisher _____
Cost $_____ Hardware _____ Date Written _____
Copy Protected? _____ Can it be listed? _____
Language _____ Need paddles, printer, etc? _____
Grade/Ability level: K 1 2 3 4 5 6 7 8 9 10 11 12
Educational Objectives? _____
_____
_____

CLASS ARRANGEMENT
_____ class demo
_____ class exploration
_____ small group
_____ individual

STRATEGY INCLUDED
_____ graphics
_____ sound
_____ pre/post test
_____ other _____

INSTRUCTIONAL TECHNIQUE (check the ones that apply)

CAI:
_____ drill and practice
_____ tutorial
_____ demonstration
_____ simulation
_____ educational game

NUMERICAL ANALYSIS:
_____ lab data analysis
_____ statistical
_____ administrative

DATA PROCESSING:
_____ grade keeping
_____ inventory
_____ scheduling

_____ WORD PROCESSING
_____ MATERIALS GENERATOR
_____ LAB INTERFACING

Circle the appropriate response (0-4) for items A-E

0	1	2	3	4
strongly disagree	disagree	undecided	agree	strongly agree

A	STUDENT OUTCOMES		
	User friendly	0 1 2 3 4	
	easy to use	0 1 2 3 4	
	interesting	0 1 2 3 4	
	objective well covered	0 1 2 3 4	
	effective compared to non-computer methods	0 1 2 3 4	
B	PEDAGOGY		
	input asked for	0 1 2 3 4	
	feedback appropriate	0 1 2 3 4	
	learning fostered	0 1 2 3 4	
	good sequence	0 1 2 3 4	
	uses scientific inquiry	0 1 2 3 4	
	provides remediation	0 1 2 3 4	
C	OPERATION		
	bug free	0 1 2 3 4	
	clear directions	0 1 2 3 4	
	effective displays	0 1 2 3 4	
	retrievable menu	0 1 2 3 4	
	easily terminated	0 1 2 3	
	allows input error	0 1 2 3	
	provides reinforcement	0 1 2 3	
D	METHOD and APPROACH		
	good documentation	0 1 2 3	
	variety learning modes	0 1 2 3	
	modifiable	0 1 2 3	
E	PROGRAM CONTENT		
	considers student ability	0 1 2 3	
	suitable	0 1 2 3	
	accurate	0 1 2 3	
	easily implemented	0 1 2 3	
F	SUGGESTIONS FOR USE IN UNIT STUDY		

BEST COPY AVAILABLE

STUDENT-COMPUTER GROUPING MODES

12e page 1

INDIVIDUAL

PREFERRED MODES OF INSTRUCTION:

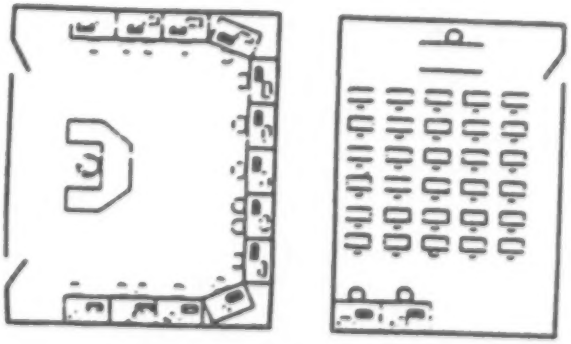
DRILL

TUTORIAL

PROBLEM SOLVING

MAIN BENEFIT:

STUDENT CONTROLS CONTENT AND PACE OF LEARNING



SMALL GROUP

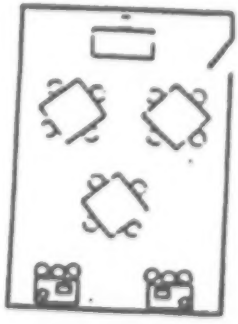
PREFERRED MODES OF INSTRUCTION:

SIMULATION

EDUCATIONAL GAME

MAIN BENEFIT:

STUDENT COOPERATION IS ENCOURAGED



Reprinted from MECC Atari Instructional Computing Demonstration Support Booklet

STUDENT-COMPUTER GROUPING MODES

13c page 2

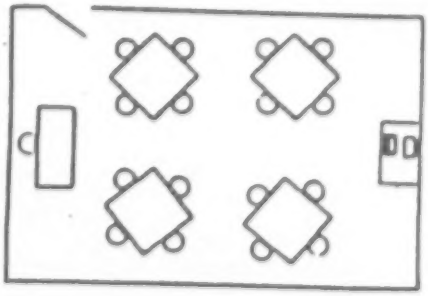
BYSTANDER

MOST LIKELY MODE OF INSTRUCTION

SIMULATION

MAIN RESULTING BENEFIT

STUDENT INTERACTION



LARGE GROUP

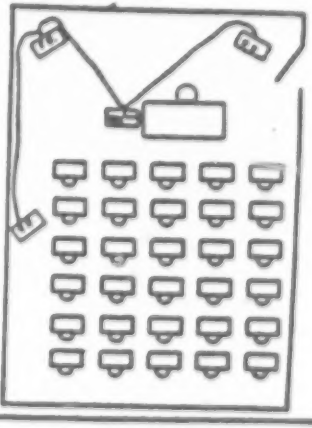
PREFERRED MODES OF INSTRUCTION:

PROBLEM SOLVING

DEMONSTRATION TOOL

MAIN BENEFIT:

TEACHER PRESENTATIONS ARE ENHANCED



Reprinted from MECC Atari Instructional Computing Demonstration Support Booklet

	Class Demonstrations	Class Explorations	Small Group Interaction	Individual Student Interactions
Drill & Practice				
Tutorial				
Simulation				
Educational Games				
Problem Solving				

Project SERAPHIM ORDER FORM

NSF Science Education; J. W. Moore and J. J. Lagowski

NOTE: *Refer to the SERAPHIM Catalog for details on each item.*

DESCRIPTION	Price		Number		Total
Written Modules					
Information Modules					
Information Sources	$ 0.25	x	_____	=	_____
Software Sources	0.25	x	_____	=	_____
Checklist for Computer Games	0.25	x	_____	=	_____
Program Contest Rules	0.25	x	_____	=	_____
SERAPHIM Catalog	1.00	x	_____	=	_____
Software List	5.00	x	_____	=	_____
Networking in Academia	1.00	x	_____	=	_____
Final Report of the 7th Int'l. Conference on Chem. Educ., Montpellier, France, August 21-26, 1983	2.00	x	_____	=	_____
Criteria for Review of Instructional Software	0.25	x		=	
Author Modules					
PETNET	0.50	x	_____	=	_____
Data Entry: A Tutorial	1.00	x	_____	=	_____
Mathematic String Evaluator	0.50	x	_____	=	_____
Using Reproducible Sequences of Random Numbers	0.50	x	_____	=	_____
Computer Simulations in Science Teaching	0.50	x	_____	=	_____
Numeric Text Formatter	0.50	x	_____	=	_____
Sample Dispenser-Grading System	2.00	x	_____	=	_____
Modification Guidelines (Bendall) (Apple #7)	4.00	x	_____	=	_____
Modification Guidelines (Bailey) (Apple #6)	4.00	x	_____	=	_____
Modification Guidelines (Kador) (Apple #13)	2.00	x	_____	=	_____
Will Computers Replace TA's? Professors? Labs? Should They?	1.00	x	_____	=	_____
POWWOW-The Future of Microcomputers in Chemistry Educ.	1.00	x	_____	=	_____
Modification Guidelines: SERAPHIM Apple Disks 1-17 distributed prior to 8/1/84	2.00	x	___	=	____
Review Modules					
Introduction to Organic Chem. (S. Smith)	1.00	x	_____	=	_____
Gas Laws (Allendoerfer)	1.00	x	_____	=	_____
Polymerlab (Williams)	1.00	x	_____	=	_____
Intro. to Polymer Chemistry (ACS)	1.00	x	_____	=	_____
Computer-Based Studies for Phys. Chem. (Barrow)	1.00	x	_____	=	_____
Molec (Owen and Currie)	1.00	x	_____	=	_____
Prelab Studies for General, Org., and Biol. Chemistry (Olmsted and Olmsted)	1.00	x	_____	=	_____
Graphical Analysis (Dice)	1.00	x	_____	=	_____
Concentrated Chemical Concepts (Cornelius)	1.00	x	_____	=	_____
Computer Aided Instruction for Gen. Chem. (Butler & Hough)	1.00	x	_____	=	_____
Titrator (Dice)	1.00	x	_____	=	_____
Gradisk (Cornelius)	1.00	x	_____	=	_____
Microchem (Oulette)	1.00	x	_____	=	_____
Instructional Text Modules					
Kinetics: A Study of Reaction Rates	0.50	x	_____	=	_____
Least Squares, Pseudorotation Parameters	4.00	x	_____	=	_____
Quantum Mechanics by Finite Difference Methods	4.00	x	_____	=	_____
Interfacing Instruments	4.00	x	_____	=	_____
RADIAL	2.00	x	_____	=	_____
Software Modules					
Software Modules: Atari (800, 48K) (Documentation included)					
Disks #1 and #2	8.00	x	_____	=	_____
			SUB-TOTALS	=	_____

BEST COPY AVAILABLE

(OVER...)

BEST COPY AVAILABLE

Project SERAPHIM ORDER FORM

DESCRIPTION	Price	Number	Total
Software Modules: IBM PC (DOS 1.0 or 1.1) (Documentation included)			
Disk #1	8.00 x	____	= ____
Disk #2	8.00 x	____	= ____
Disk #3	8.00 x	____	= ____
Project CHEMLAB: single-sided (160K) drives on IBM PC; 9 disks	36.00 x	____	= ____
Project CHEMLAB: double-sided (320K) drives on IBM PC; 5 disks	20.00 x	____	= ____
Software Modules: Commodore PET/C-64 (4000 Series, BASIC 3.0) (Documentation included)			
8050 Disk #1	$ 8.00 x	____	= ____
Cassette #1 and #2	5.00 x	____	= ____
Cassette #3	3.00 x	____	= ____
C-64 Disk #1	8.00 x	____	= ____
Software Modules: Radio Shack TRS-80 (TRS DOS 1.3) (Documentation included)			
Disk #1	8.00 x	____	= ____
Disk #2	8.00 x	____	= ____
Disk #3 (in Spanish)	8.00 x	____	= ____
Software Modules: Apple II (DOS 3.3) (Documentation included)			
Disks #1 and #2	8.00 x	____	= ____
Disks #3 and #4	8.00 x	____	= ____
Disk #5	8.00 x	____	= ____
Disk #6	8.00 x	____	= ____
Disk #7	8.00 x	____	= ____
Disk #8 * See also Modification Guidelines for SERAPHIM Apple Disks 1-17 distributed prior to 8/1/84; on reverse side.	8.00 x	____	= ____
Disk #9	8.00 x	____	= ____
Disk #10	8.00 x	____	= ____
Disk #11	8.00 x	____	= ____
Disk #12	8.00 x	____	= ____
Disk #13	8.00 x	____	= ____
Disk #14	8.00 x	____	= ____
Disk #15	8.00 x	____	= ____
Disk #16	8.00 x	____	= ____
Disk #17	8.00 x	____	= ____
Disk #18	8.00 x	____	= ____
Disk #19	8.00 x	____	= ____
Disk #20	8.00 x	____	= ____
SUB-TOTAL THIS PAGE		____	= ____
SUB-TOTAL FIRST PAGE		____	= ____
TOTAL		____	= ____

NOTE: Payment must accompany all orders. Make checks payable to: Project SERAPHIM.

Your name: _____

Address: _____

_____

_____

RETURN THIS FORM TO: Dr. John W. Moore, Director; Project SERAPHIM; Department of Chemistry; Eastern Michigan University; Ypsilanti, MI 48197; (313) 487-0368; 487-0186.

THE RIB STRIKES BACK:
COUNTERING GENDER INEQUITIES IN COMPUTER USE

Dr. Danna Fertsch
College of Education
University of Houston
Houston, Texas

James R. Aman
St. Pius X High School
811 Donovan
Houston, Texas 77091

Abstract:
The presence of gender inequities in computer-related courses has received increasing attention in recent months. This presentation discusses specific techniques for overcoming existing biases and practical strategies for preventing their appearance.

Are computer-related courses and activities geared more for males than for females? Are females denied the right to learn and to practice computer skills? Unfortunately, the answer to both questions is "Yes!" The body of research identifying the existence of gender bias in computer classes grows each month. As educators, we must not only be aware of the existence of the bias, but also prepared to take corrective action. The practice of computer education must be improved immediately. The identification and implementation of specific strategies is the first step in this direction. Whether it can be done successfully depends upon educators realizing that gender biases exist, and that they are part of the problem as well as part of the solution.

Here are twenty-three specific strategies and techniques for countering gender bias.

1. School administration, faculty, and staff need to be made aware of the gender-gap problem. Their input and assistance in identifying solutions should be sought.
2. Educators should develop and actively participate in a support network with other educators attempting to provide equal computer education to all students.
3. Parent computer education sessions need to be conducted. The focus of these sessions would be on: 1) providing information about the uses of computers in homes and schools, 2) providing opportunities for "hands-on" experiences to reduce anxiety, 3) stressing the necessity of computer education for all students, regardless of gender, and 4) urging equal access to and encouragement in the use of any home computer. Discussion of gender inequities would focus parents' attention on whether all children in a household are equally encouraged to use the computer.
4. During computer-oriented teacher inservices, brainstorming could be done on ways that teachers could recognize and combat gender discrimination and role stereotyping.
5. Teachers are powerful role models for their students. By their actions, they communicate standards of gender-appropriate behavior. Teachers need to be aware of their impact and make a conscious effort not to discourage female students from computer study. They need to train themselves to pay as much attention to female students as they do to males and to not ignore female students' questions and concerns. When students are ignored in a class, they may begin to believe that knowledge of the subject matter is not

important or necessary for them. This situation has occurred within mathematics classes, and could do so within computer classes or laboratories, if teachers are not watchful.

6. Educators need to be aware of gender biases in commercial software. They need to carefully evaluate all software for gender-appropriateness. Much marketed software is male-oriented, both in content and reinforcers, and is unappealing to females. Having students participate in this process can make them aware of their own and the software's biases.

7. Computer use should be integrated into all areas of the school curriculum, especially those viewed as non-gender-specific by females. Use of the computer within social studies, art, and English classes can expose females to non-mathematical computer uses.

8. Extensive opportunities for non-programming computer experiences should be incorporated into the curriculum and classroom. If females can use computers in a variety of "useful" ways, i.e., word processing, file management, and electronic spreadsheets, they could become more enthusiastic about computer use and enroll in additional computer courses.

9. Computers should be used within classrooms and not just within a laboratory setting. The laboratory may be viewed as "male turf" by females, and they may not feel comfortable using the computer there. However, once the computer appears in the classroom, it becomes common ground.

10. Students could be assigned research projects on different computer career/occupational fields. Class discussions

related to the assignment could include the gender-appropriateness of each career field and data from current computer career salary surveys.

11. Female speakers from business, industry, and education should be invited to address students about computer use in their respective fields. They could not only furnish information about computer-related occupations, but could also become role models for the female students.
12. The study of computer history should include women in computing. Students need to be made aware that persons of both genders contribute to technological development.
13. In addition to programming projects of a non-gender-specific nature, additional gender-specific projects should be assigned to whole classes. These additions would increase cross-gender awareness and sensitivity.
14. The computer laboratory or classroom should be physically arranged to facilitate student interaction and group activities. Adolescent females are very social, and isolated, individual activities do not meet their social interaction needs. Even older females prefer to work with friends at the computer. Unless social (human) interaction is allowed during computer activities, females could view the computer as cold and inhuman, an experience to be avoided.
15. To ensure extracurricular use of the computers by female students, special "female only" computer sessions could be initiated. Alternatively, half of the computers could be

reserved for females. However, mixed-gender extracurricular computer sessions would have to be closely monitored to ensure that females have equal access and were not pressured out by males.

16. Because adolescent females are oriented toward social interaction, they will be more likely to actively use the computer laboratory if their friends are there, also. Therefore, key groups of females should be "targeted" and encouraged as computer users. Once some females begin to be visible in the laboratory, others will come, too.
17. Females should be encouraged to actively participate in an extracurricular Computer Club or Student Computer Specialist group. Functions of this group could include peer-tutoring, software evaluation, and creation of software for use elsewhere in the school. Females in this group would have visibility and serve as peer role models for other females.
18. Female computer students should act as demonstrators and instructors at parent education sessions. Female student participation could create a visible association between females and computers for both parents and students. Additionally, the students would serve as peer role models for each other.
19. Females having computers at home must be encouraged to actively use them and not be pushed aside by male siblings.
20. Teachers recommending students for summer computer camps and classes should encourage female as well as male attendance. There is some speculation that the recommendation process is currently (undoubtedly unconsciously) biased toward male

Mark Sullivan
La Marque I.S.D.
1727 Bayou
La Marque, TX 77568

THE HOME CONNECTION
-helping families use computer tools

Introduction

This paper offers some general directions for developing a computer literacy curriculum that helps students use home computers more effectively. If you would like more specific information about the skills and strategies involved, write me in care of the school district address above. While you're at it, send me some of your thoughts on the matter.

Philosophically, this curriculum is based on the teachings of John Dewey and the other educational pragmatists. It stresses meeting the present and real-world educational needs of the learner. Unlike the more classical approaches that presently seem to dominate our schools, this curriculum starts with the student, not the subject matter. In my opinion, pragmatic curricula are not just the best way to teach computer literacy, they are the only way. It seems futile to try to "prepare" students for a world with computers when no one really knows what that world will be like. The result of such attempts is often a vague curriculum with contrived and arbitrary skills.

To be truly pragmatic, a computer literacy curriculum must also be dynamic. A rapidly changing technology means rapidly changing educational needs. Right now, the biggest educational deficit the general population has about computers concerns the use of home computers. Many of our students have computers in their homes, but few of them are using them effectively. So, in developing a curriculum, I started with this question: "What do we need to teach people to help them use home computers better?"

I found that the skills could be sorted into six categories:

1) Computer Consumer Skills
2) Microcomputer Operation and Care
3) Keyboarding Skills
4) Familiarity with the Scope of Home Software
5) Use of Standard Productivity Software
6) Developing Technological Support Systems

I have limited the scope of this curriculum to include only those skills needed to successfully use home microcomputers. For that reason, many "typical" computer literacy skills are not covered. For instance, I have not included a programming strand because people can use computers quite effectively without knowing how to program. If you like this pragmatic approach, but still want to offer programming skills, go ahead. Be eclectic! There are many fine programming curricula that can be merged with this one.

At the end of this paper is a section about notebook computing. Few students are presently toting notebook computers around school, but that will be changing very soon. We need to start planning for that eventuality, because it will drastically change the face of home

and educational computing. I offer a few thoughts on the matter for your consideration.

Consumer Skills for Computers

I listed this educational need first because the lack of computer consumer skills is the biggest obstacle in the way of successful home computing. Many people want to take advantage of computers, but are understandably reluctant to spend hundreds of dollars for a device they know little about. Worse yet, many people are willing to pay the money, but then become disillusioned because they didn't get what they wanted. In the end, our students suffer for their parents' lack of knowledge.

I would like to see the public schools develop innovative adult education programs to alleviate this problem. We need to teach parents the process of configuring and purchasing a computer system. Also, we need to help them obtain the information they need to make intelligent choices. When properly guided, even families with modest incomes can afford to give their children the challange and advantage of a computer at home.

Microcomputer Operation and Care

These skills are the simple but important things the students need to learn in order to start using computers: how to turn the machine on, how to handle diskettes, how to load and run programs, etc. Conventional computer literacy curricula teach only those skills needed to use microcomputers in a school environment. Home computer use requires additional skills. For example, most home applications require knowledge about printers. We need to teach students how to load paper and how to change ribbons. Also, when students use computers at home, they must maintain their own tape or diskette library. That means they need to know how to organize and back-up their tapes or diskettes. Skills like these aren't usually required of a student in school applications, so they aren't usually taught.

Teaching the fundamentals of computing takes more skill than teaching more advanced topics does. It is at this point that, for some students, computer anxiety interferes with learning. A student-centered teaching style can help. We, as teachers, should not overly concern ourselves with the educational objectives at this point. Rather, we should focus upon our own behavior, and upon the interaction that is occuring between the student and the machine. Computer anxiety is a very real thing for many students, but with sensitivity and care it can be overcome. If students can be helped to feel masterful and competent, the sense of awe and fear they have about computers can be transformed into a sense of marvel and magic!

Familiarity with the Scope of Home Software

We should encourage our students to use a variety of software, just as we encourage them to read a variety of books. There are so many good things out there: strategy games, interactive fiction, simulations, word processors, computer art, data bases, music...the list goes on and gets longer every day! Even an occasional game of "Astrosmash" can be indulged in without too much brain damage. These

programs belong in our classrooms and libraries, because they are an important part of the literature of the 1980's.

Use of Standard Productivity Software

Productivity programs are the basic tools of computing: word processors, data bases, spreadsheets, and graphic packages. They are frequently "integrated" into one program (e.g. Lotus 1-2-3) so that data can be passed from one tool to the next without retyping. They were originally devised with business people in mind, but they also ideal for students or anyone else who needs to gather and manipulate information.

The primary way young people can use productivity software is in helping them with their schoolwork. For that reason, and in keeping with the pragmatic approach, productivity software skills should be taught in conjunction with regular scholastic skills. We should teach "computing in the subject matter" just as we now teach "reading in the subject matter".

Developing Technological Support Systems

Using a computer involves overcoming problems. The main thing that seperates successful home computer users from frustrated home computer users is that the successful users know where to go to get help when they need it. Through the use of friends, magazines, users' groups, and modems, successful users "network" themselves into larger "computing communities". The schools can help students become involved in this process, and can themselves become part of the process. (For example, for $150 in hardware and software, a district could set up its bulletin board system that could be run after school on existing phone lines and maintained during school as a class project.) A one-time computer literacy course is not enough. We need to help students develop an environment of continued learning.

Notebook Computers

Notebook computers are powerful personal computers that run on batteries and have special built-in flat display screens. They are called "notebook computers" because they are the size and weight of a typical three-ring notebook and because they can be used to do much the same thing, only better. Notebook computers usually have productivity software in ROM. (That is to say, the software is available as soon as the computer is turned on.) Also, programs can be loaded from a cassette or floppy disk. Many of these computers can even be loaded with more than one program at a time, and the user can switch between them instantaneously.

What can students do with notebook computers? As the name implies, their most obvious application is in taking notes. When in class or at the library, students can input notes directly into a word processor. Then, whenever they have a few minutes to spare, they can rewrite their notes. If they need reports, they can write and edit them in their spare time directly from their notes. When they get home, they can plug their computer into their printer. Three minutes later, they are outside playing baseball or having fun with their friends.

Many notebook computers have clock/calendar functions for

keeping track of appointments and assignments. Some have built in modems that can be used to access research data bases. And, of course, educational software can be run on them whenever it is convenient for the student.

Notebook computers put computing power at the students' fingertips whenever they need it. That is a giant step foward toward a truly useful computer. Right now, the reason we don't see more of them in the schools is price. Good notebook computers start at about $500. That's reasonable by microcomputer standards, but it's a lot of money to tuck under your kid's arm in the morning! Still, the price of computers is determined almost entirely by demand. Notebook computers are such enabling tools that the demand is sure to go up and the price is sure to come down. We need to be ready for it when it does.

THE COMPUTER IN THE CHRISTIAN ENVIRONMENT

Bill Cobb, Second Baptist School

Computers are used as useful tools in many new and exciting ways. Over twenty-five percent of all the families in America have computers in their homes and this number is growing rapidly as this technology continues its growth and decreasing "price tag."

It is our belief that God is the Creator of the laws of technology by which computers function. Man has made progress in the field of computers only by the discovery of and obedience to these Laws of God.

We also believe that God's laws are to be interpreted in a physical as well as a spiritual sense in today's world. It is expected that we use good stewardship in promoting the Good News of salvation in an urgent and accurate way. Computers are fast and accurate tools of communication. This places a direct challenge to the Christian schools. With the assistance of computers, we can communicate the Good News of salvation as accurately and as urgently as possible.

At the present time there is a void of Bible-centered computer software. There is great potential for teaching Christian values to all children both at home, school, and at church with the aid of the computer.

Learning spiritual truths through the memorizing of Bible verses and Books of the Bible as well as computer games can give new dimensions to a field which can be usually thought of as being "dry, rote learning". The promise of immediate rewards reinforced by repetition teaches knowledge, and the value of that knowledge depends upon the sources used.

Other examples of uses Christians can make of computers are: translation comparisons (through word processing), evangelistic letter writing, church committee reports, prayer requests and answered prayers, Christmas mailings, church financial and attendance records, family membership directory data, Biblical research and many more.

RESOURCES, RESOURCES, RESOURCES! HOW DO I CHOOSE?

In an attempt to assist those teachers who have Apple computers in their classrooms, I have compiled a list of books that might be used as reference materials in your classroom. A list of computer magazines is also included.

I will bring copies of these books and magazines to the resource session and will attempt to show books useful for specific areas such as BASIC, music, graphics, computer literacy, etc.

Teachers should also be aware of public domain microcomputer software available for specific computers and plan to utilize this in the classroom. Region IV Education Service Center has an extensive library of Apple and Commodore public domain software which might prove useful to you and your students.

Marilyn T. Fricks, Coordinator
Computer Based Instruction
Region IV Education Service Center
P.O. Box 863
Houston, TX 77001

1 RESOURCES! RESOURCES! RESOURCES! 05/85

TITLE	AUTHOR	PUBLISHER
* 1001 THINGS TO DO W/THE MACINTOSH	SAWUSCH AND SUMMERS	TAB BOOKS INC.
* 101 WAYS TO USE A MACINTOSH	DAVID D. THORNBURG	RANDOM HOUSE
* BECOMING A MAC ARTIST	WAKE GUZELIMIAN	COMPUTE! BOOKS PUBLICATION
* HOW TO USE THE MACINTOSH	JERRY WILLIS	DILITHIUM PRESS
* INTRODUCING THE APPLE MACINTOSH	CONNOLLY AND LIEBERMAN	HOWARD W. SAMS AND CO., INC.
* MAC WORK MAC PLAY	LON POOLE	MICROSOFT PRESS
* MACTELECOMMUNICATIONS	ERICKSON AND CRAMER	OSBORNE MC GRAW-HILL
* THE APPLE MACINTOSH BOOK	CARY LU	MICROSOFT PRESS
* THE APPLE MACINTOSH PRIMER	WILLIAM B. SANDERS	DATAMOST INC
* THE EASY GUIDE TO YOUR MACINTOSH	JOSEPH CAGGIANO	SYBEX COMPUTER BOOKS
* USING & PROGRAMMING THE MACINTOSH	FREDERICK HOLTZ	TAB BOOKS INC.
* USING MACWRITE AND MACPAINT	TIM FIELD	OSBORNE MC GRAW-HILL
32 BASIC PROGRAMS FOR THE APPLE	RUGG AND FELDMAN	DILITHIUM PRESS
33 NEW APPLE COMPUTER PROGRAMS	WHITE, FRED	ARCSOFT PUBLISHERS
A BIT OF BASIC	DWYER AND CRITCHFIELD	ADDISON-WESLEY PUBLISHING CO
A GUIDE TO PROGRAMMING IN APPLESOFT	PRESLEY, BRUCE	VAN NOSTRAND REINHOLD
A NEW TRUE BOOK COMPUTER	KAREN JACOBSEN	REGENSTEINER PUBLISHING ENTER
ACCOUNTANT'S BASIC PROGRAMMING	PARKER & STEWART	RESTON PUBLISHING COMPANY, INC
AN APPLE FOR THE TEACHER	CULP & NICKLES	BROOKS/COLE PUBLISHING CO
AN INTRO TO MICROCOMPUTERS VOLUME 0	OSBORNE & BUNNELL	OSBORNE/MCGRAW-HILL
ANIMATION, GAMES, AND SOUND (APPLE)	FABBRI,TONY	PRENTICE-HALL, INC.
APPLE ASSEMBLY LANGUAGE	MAURER, W. DOUGLAS	COMPUTER SCIENCE PRESS
APPLE BACKPACK	KAMINS & WAITE	BYTE BOOKS
APPLE BASIC	HASKELL, RICHARD	PRENTICE-HALL
APPLE BASIC FOR BUSINESS	PARKER & STEWART	RESTON PUBLISHING CO
APPLE BASIC: DATA FILE PROGRAMMING	FINKEL & BROWN	JOHN WILEY & SONS, INC
APPLE FILES	MILLER, DAVID	RESTON PUBLISHING CO
APPLE GRAPHICS	BAILEY AND KERLIN	ROBERT J. BRADY CO
APPLE GRAPHICS GAMES	COLETTA, PAUL	RESTON PUBLISHING CO
APPLE GRAPHICS/ARCADE GAMES DESIGN	STANTON, JEFFREY	THE BOOK COMPANY
APPLE II ASSEMBLY LANG EXERCISES	SCANLON, LEO J	JOHN WILEY & SONS, INC
APPLE II COMPUTER GRAPHICS	WILLIAMS, KEN, ET AL	ROBERT J. BRADY CO
APPLE II DISK GUIDE	GIFFORD, ZELDA	OSBORNE/MCGRAW-HILL
APPLE II PROGRAMMER'S HANDBOOK	VILE, RICHARD C	PRENTICE-HALL
APPLE II USER'S GUIDE	POOLE, LON	OSBORNE/MCGRAW-HILL
APPLE II WORD PROCESSING	POLING, CAROL	QUE CORPORATION
APPLE II-6502 ASSEMBLY LANG TUTOR	HASKELL, RICHARD	PRENTICE-HALL
APPLE LOGO	ABELSON, HAROLD	BYTE/MCGRAW-HILL
APPLE MACHINE LANGUAGE	INMAN & INMAN	RESTON PUBLISHING CO
APPLE MACHINE LANGUAGE/ROSEN	ROSEN, ROBERT D	HOLT, RINEHART AND WINSTON
APPLE PASCAL A HANDS-ON APPROACH	LUEHRMANN AND PECKHAM	MCGRAW-HILL
APPLE PASCAL GAMES	HERGERT AND KALASH	SYBEX
APPLE SOFTWARE 1984	STANTON, ET AL (EDS)	THE BOOK COMPANY
APPLE SOFTWARE BANK VOL 3-5		APPLE COMPUTER INC
APPLE SOFTWARE BANK VOLS 1 & 2		APPLE COMPUTER INC
APPLESOFT BASIC FOR APPLE II & IIE	GRAFF & GOLDSTEIN	PRENTICE-HALL/COMMUNICATIONS
APPLIED APPLE GRAPHICS	PIP FORER	PRENTICE-HALL INC
ART & GRAPHICS ON THE APPLE II/IIE	WILLIAM H DEWITT	JOHN WILEY & SONS, INC
ASSEMBLY LANG PROGRAMMING FOR APPLE	MOTTOLA, ROBERT	OSBORNE/MCGRAW-HILL

TITLE	AUTHOR	PUBLISHER
ASSEMBLY LINES: THE BOOK	WAGNER, ROGER	SOFTALK PUBLISHING
BASIC A HANDS-ON METHOD	PECKHAM, HERBERT D.	MCGRAW-HILL
BASIC BEGINNINGS	LIPSCOMB & ZUANICH	AVON CAMELOT BOOKS
BASIC BUSINESS BASIC USING MICROS	MEARS & RAHO	BROOKS/COLE PUBLISHING CO
BASIC COMPUTER PROGS FOR THE HOME	STERNBERG, CHARLES D.	HAYDEN BOOK CO
BASIC EXERCISES FOR THE APPLE	LAMOITIER, JEAN-PIERRE	SYBEX
BASIC FOR THE APPLE II	GOLDSTEIN & GOLDSTEIN	ROBERT J BRADY CO
BASIC/APPLE IIE A PROG. GUIDE	TUCKER, ALLEN B. JR	HOLT, RINEHART AND WINSTON
BEGINNING COMPUTER PROGRAMMING	FRENCH, ET AL	ALLYN AND BACON
BETTER BASIC FOR THE APPLE	HUME & HOLT	RESTON PUBLISHING CO
CAI SOURCEBOOK	ROBERT L. BURKE	PRENTICE-HALL, INC
COMPUTER LITERACY/HANDS-ON APPROACH	LUEHRMANN & PECKHAM	MCGRAW-HILL
COMPUTERS & DATA PROCESSING W/BASIC	STEVEN L. MANDELL	WEST PUBLISHING CO
COMPUTERS FOR EVERYBODY	WILLIS & MILLER	DILITHIUM PRESS
COMPUTERS FOR KIDS APPLE II EDITION	LARSEN, SALLY GREENWOOD	CREATIVE COMPUTING PRESS
COMPUTERS IN THE SCHOOLS		THE HAWORTH PRESS INC
CRASH COURSE IN MICROCOMPUTERS	FRENZEL, LOUIS E. JR	HOWARD W. SAMS
DEVELOPING COMPETENCY-BASED TRAIN	WILLIAM E. BLANK	PRENTICE-HALL, INC
DISCOVERING APPLE LOGO	THORNBURG, DAVID D	ADDISON-WESLEY PUBLISHING CO
DON'T (HOW TO CARE FOR YOUR COMP.)	ZAKS, RODNAY	SYBEX
EDUCATIONAL SOFTWARE DIRECTORY		STERLING SWIFT
FANCY PROGRAMMING IN APPLESOFT	CUELLAR, GABRIEL	RESTON PUBLISHING CO
FOR ALL AGES COMPUTER TOOLS		SPINDEX CORPORATION
GETTING DOWN TO BASIC	HANSEN & BINGHAM	K-12 MICROMEDIA
GLOSSARY OF COMPUTER TERMS	JOHN PRENIS	RUNNING PRESS
GOLDEN DELICIOUS GAMES FOR APPLE	FRANKLIN, ET AL	JOHN WILEY & SONS, INC
GUIDEBOOK TO SMALL COMPUTERS	WILLIAM BARDEN, JR	HOWARD W. SAMS & CO. INC
HI-RES SECRETS	DON FUDGE	AVANT-GARDE CREATIONS
HOW TO BUY A WORD PROCESSOR (GUIDE)	SCRIVEN & MANUS	ALFRED PUBLISHING CO. INC
I SPEAK BASIC TO MY APPLE	JONES, AUBREY B, JR	HAYDEN BOOK CO
INTERMEDIATE-LEVEL APPLE HANDBOOK	HEISERMAN, DAVID L	HOWARD W. SAMS
INTRO TO COMPUTERS & BASIC PROGRAM	BRENAN & MANDELL	WEST PUBLISHING CO
INTRO TO COMPUTERS AND BASIC PROG	BRENAN & MANDELL	WEST PUBLISHING CO
INTRO TO PASCAL/STRUCTURED DESIGN	DALE & ORSCHALICK	D C HEATH AND COMPANY
INTRODUCTION TO BASIC PROGRAMMING	SHELLY & CASHMAN	ANAHEIM PUBLISHING CO
INTRODUCTION TO COMPUTER ANIMATION	WADSWORTH, NAT	HAYDEN BOOK CO
INTRODUCTION TO COMPUTER MUSIC	WAYNE A BATEMAN	JOHN WILEY & SONS
INTRODUCTION TO PASCAL	NEILL GRAHAM	WEST PUBLISHING CO
KAREL THE ROBOT	PATTIS, RICHARD E	JOHN WILEY & SONS, INC
KATIE AND THE COMPUTERR	D'IGNAZIO, FRED	CREATIVE COMPUTING PRESS
KIDS AND THE APPLE	CARLSON, EDWARD H	DATAMOST (RESTON)
KILOBYTE KID'S - PERSONAL COMPUTERS	SPANGENBURG & MOSER	WADSWORTH ELECTRONIC
LEARNING LOGO ON THE APPLE II	MCDOUGALL/ADAMS/ADAMS	PRENTICE-HALL OF AUSTRALIA
LEARNING WITH LOGO	WATT, DANIEL	MCGRAW-HILL
LOGO FOR THE APPLE II	ABELSON, HAROLD	BYTE/MCGRAW-HILL
MECC TRAINING MANUALS (SEVERAL)	MECC	MECC
MICRO ON THE APPLE, VOL 1	CAVALLARI, FORD (ED)	MICRO INK, INC
MICRO ON THE APPLE, VOL 2	CAVALLARI, FORD (ED)	MICRO INK, INC
MICROCOMPUTER GRAPHICS	MYERS, ROY E	ADDISON-WESLEY PUBLISHING CO
MICROCOMPUTER WORKBOOK (APPLE ED)	POIROT, JAMES, ET AL	STERLING SWIFT

TITLE	AUTHOR	PUBLISHER
MINDSTORMS	PAPERT, SEYMOUR	BASIC BOOKS, INC
MINUTE MANUAL FOR APPLE WRITER II/E	PIRISINO, JIM	MINUTEWARE
MINUTE MANUAL FOR APPLEWRITER II+	JIM PIRISINO	PIRISINO & MINUTEWARE
MINUTE MANUAL FOR APPLEWRITER IIE	JIM PIRISINO	PERISINO & MINUTEWARE
MOSTLY BASIC: APPLIC FOR YOUR APPLE	BERENBON, HOWARD	HOWARD W SAMS
PASCAL	DALE & ORSHALICK	D C HEATH AND CO
PASCAL FOR THE APPLE	MACCALLUM, IAIN	PRENTICE-HALL
PASCAL PROGRAMMER		THE DENVER SOFTWARE CO
PASCAL PROGRAMMING FOR THE APPLE	LEWIS, T.G.	RESTON PUBLISHING CO
PASCAL TUTOR		THE DENVER SOFTWARE CO
PEANUT BUTTER & JELLY GUIDE TO COMP	WILLIS & MILLER	DILITHIUM PRESS
PERSONAL GRAPHICS	BARNETT & BARNETT	LITTLE, BROWN AND COMPANY
PICTURE THIS! INTRO TO GRAPHICS	DAVID D THORNBURG	ADDISON-WESLEY PUBLISHING CO
POCKET GUIDE TO PASCAL	WATT, DAVID	ADDISON-WESLEY PUBLISHING CO
PRACTICAL GUIDE TO COMPUTERS IN ED	COBURN ET AL	ADDISON-WESLEY PUBLISHING CO
PRACTICAL PASCAL	GILBERT & LARKY	SOUTH-WESTERN PUBLISHING CO
PREPARING INSTRUCTIONAL OBJECTIVES	ROBERT F MAGER	FEARON PUBLISHERS
PROGRAMMING THE 6502	RODNEY ZAKS	SYBEX INC
PROGRAMMING THE APPLE II	CAMPBELL & ZIMMERMAN	MESA RESEARCH
PROGRAMMING THE APPLE II IN BASIC	TEBBE, PAUL	PRENTICE-HALL
RUN: COMPUTER EDUCATION	HARPER & STEWART	BROOKS/COLE PUBLISHING CO
SHARON'S BEGINNERS COMPUTER BOOK		SHARON PUBLICATIONS INC
SWIFT'S EDUCATIONAL SOFTWARE DIRECT		STERLING SWIFT PUBLISHING CO
TECHNIQUES OF BASIC	GRILLO & ROBERTSON	WM C BROWN CO
TEN STATEMENT FORTRAN + FORTRAN IV	KENNEDY & SOLOMON	PRENTICE-HALL INC
THE 3-D ANIMATED APPLE	PHILIP COHEN	PRENTICE-HALL OF AUSTRALIA PTY
THE APPLE ALMANAC	GOEZ & SANDERS	DATAMOST
THE APPLE PERSONAL COMP FOR BEG	DUNN & MORGAN	PRENTICE-HALL
THE APPLE USER'S ENCYCLOPEDIA	PHILLIPS/CONKLIN/SCELLATO	THE BOOK COMPANY (ARRAYS)
THE BASIC CONVERSIONS HANDBOOK	BRAIN/OVIATE/PAQUIN/STONE	HAYDEN BOOK COMPANY INC
THE BEGINNER'S COMPUTER DICTIONARY	WALL AND WALL	AVON CAMELOT
THE BEST OF MICRO VOLUME 3		MICRO INK, INC
THE BYTE BOOK OF PASCAL	LIFFICK, BLAISE W (ED)	BYTE/MCGRAW-HILL
THE COMPUTER IN THE SCHOOL	TAYLOR, ROBERT P (ED.)	TEACHERS COLLEGE PRESS
THE COMPUTER TUTOR	ORWIG, GARY W.	WINTHROP PUBLISHERS
THE COMPUTER TUTOR LEARNING ACT	ORWIG & HODGES	WINTHROP PUBLISHERS, INC
THE CREATIVE APPLE	PELCZARSKI AND TATE (EDS)	CREATIVE COMPUTING PRESS
THE ELEMENTARY APPLE	SANDERS, WILLIAM B	DATAMOST
THE FOURTH LEG OF THE APPLE		BRINKER COMPUTING
THE LITTLE BOOK OF BASIC STYLE	NEVISON, JOHN M	ADDISON-WESLEY PUBLISHING CO
THE MIND TOOL (INSTRUCTOR MANUAL)	NEILL GRAHAM	WEST PUBLISHING CO
THE MIND TOOL (SOCIETY) 3RD EDITION	NEILL GRAHAM	WEST PUBLISHING CO
THE MIND TOOL - COMPUTER & SOCIETY	NEILL GRAHAM	WEST PUBLISHING CO
THE POWER OF VISICALC	WILLIAMS & TAYLOR	MANAGEMENT INFORMATION SOURCE
THE POWER OF VISICALC VOLUME II	WILLIAMS & KING	MANAGEMENT INFORMATION SOURCE
THE VISIBLE COMPUTER: 6502	ANDERSON, CHARLES	SOFTWARE MASTERS
UNDERSTANDING APPLE GRAPHICS	PEDDICORD, RICHARD	ALFRED PUBLISHING CO, INC
UNDERSTANDING BASIC (GUIDE)	PEDDICORD, RICHARD G.	ALFRED PUBLISHING CO, INC
UNDERSTANDING FORTRAN	MICHEL H BOILLOT	WEST PUBLISHING CO
USING 6502 ASSEMBLY LANGUAGE	HYDE, RANDY	DATAMOST

BEST COPY AVAILABLE

TITLE	AUTHOR	PUBLISHER
USING COMPUTERS TO ENHANCE TEACHING	SADOWSKI & LOVETT	UNIVERSITY OF HOUSTON
USING VISICALC GET DOWN TO BUSINESS	KLITZNER & PLOCIAK, JR	JOHN WILEY & SONS, INC
VISICALC HOME AND OFFICE COMPANION	CASTLEWITZ, ET AL	OSBORNE/MCGRAW-HILL
WHAT'S WHERE IN THE APPLE	LUEBBERT, WILLIAM F	MICRO INK, INC
WILL ..TELL ME WHAT AN APPLE CAN DO	POLIN, GLENN M. (ED)	STERLING SWIFT
WORD PROCESSING PRIMER	WAITE & ARCA	BYTE/MCGRAW-HILL
WRITE YOUR OWN APPLE GAMES	ANSTIS, STUART	CREATIVE COMPUTING PRESS
YOUR APPLE II NEEDS YOU	FRANK WATTENBERG	PRENTICE-HALL INC

Teaching Junior High Computer Literacy

The Texas Basic In The Summer

Teaching Jr. High computeracy in the summer involves cramming a lot of hands-on activities in a short span of days. It also means having a motivated group working in three concentrated hour-long sessions per day. Our summer school this year will be divided into three sessions per day with short breaks scheduled between each including one-half hour for lunch.

The plan is to use the first session for individual student discoveries using short programs on task cards. The typical lesson on the task cards will be leading into the lesson to be presented at large group time. The students will be typing these in on the computers, running the program, and recording output in their notebooks. Other students may use this time learning computer vocabulary while exploring the learning centers.

The middle session, or large group time, will contain the main lessons for the day. At the beginning, the use of lectures with overhead projector transparencies, posters, computer demonstrations, will fill this slot. After several days, groups will be reporting, panels of students will be discussing social issues, and films or videos will be viewed.

The last session will be either library time for research on data base projects, or time to explore software available and learn about LOGO. Those students that used the learning centers during morning session may now be having hands-on time with the other group now flip-flopping to the learning centers.

CLASS MANAGEMENT

Even though summer students are self-motivated, the early morning hours plus the days after the newness has worn off call for special classroom management ideas. Students are assigned to computers and are aware of the time allotment. Program assignments are specific, written on the chalkboard, and at the student's pace.

On the first day rules are posted on the wall. Copies are distributed with instructions to have them signed by parents and brought back.

A wall poster of historical highlights in computing serves as a motivator to students. A photo of each student is placed on the poster and advanced as each progresses through programming. At the end of the session awards are given for each top student.

Slower students may copy written programs with the understanding that they learn from each program by reading. With all the stress on equity, the intent of the course is to develop a sense of control of the computer at the student's skill level.

LOGO Olympics also brings in different levels of learning with students practicing directing the turtle through transparencies of track and field events. The last days of session students compete with each other for ribbons and awards designed by them.

After color graphics lessons, artistic creativity is stirred with an art contest with judging by school personnel and parents.

Parents are involved in the session also with daily notes written by groups of students on the word processor listing skills learned.

A hard copy of the best programs developed by the class will be kept in a scrap book to be sent to Austin as requested by TEA official Sandy Pratcher at the TCEA Annual conference.

OUTLINE OF TOPICS IN PRESENTATION

A. COMPUTER-RELATED TERMINOLOGY AND USE
   1. COMPUTER CLIMB GAME
   2. POPCORN PROCESSING
   3. I HAVE / WHO HAS GAME
   4. COMPUTER TRIVIA GAME
   5. MICRO-TIMES
B. HISTORY AND DEVELOPMENT OF COMPUTERS
   1. FILMSTRIPS
   2. COMPUTER TRIVIA GAME
   3. MICRO-TIMES
C. USE OF COMPUTER AS A TOOL
   1. STUDENT INFORMATION DATABASE
   2. NUITRITION DATABASE
   3. LETTERS TO PARENTS ON WORD PROCESSOR
   4. MESSAGE DISK BY WORD PROCESSING
   5. WRITERS DOZEN IDEAS ON WORD PROCESSOR
D. COMMUNICATE INSTRUCTIONS TO THE COMPUTER
   1. TASK CARDS
   2. PROGRAM OF THE MONTH FOLDERS
   3. GRAPHICS USING COUNTED CROSS STITCH DESIGNS
   4. LOGO
E. PROBLEMS AND ISSUES OF COMPUTER USE IN THE SOCIETY
   1. CLIP AND LAMINATE CARTOONS
   2. USING CARTOONS WITH CAPTIONS REMOVED
   3. USING VIDEO "WAR GAMES"
   4. USING "COMPUTER TRESPASSERS"
   5. AFTER DATABASE ACTIVITIES DISCUSS PRIVACY ISSUES
F. SELECTED ACTIVITIES
   1. KEYBOARD COMPANION
   2. MAKE A DATA BANK
   3. LOGO NOTEBOOK
G. AWARDS DAY
F. EVALUATION

RESOURCES

TEACHING AND COMPUTERS
CLASSROOM COMPUTER LEARNING
ELECTRONIC LEARNING
ARITHMETIC TEACHER
TEA COMPUTER LITERACY CURRICULUM GUIDE